Virtual Organization Process Integration and Execution Through Grid Computing.

THESIS

PRESENTED AS A PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF:

Master of Science in Information Technology

BY:

Juan Paulo de Anda González

Monterrey, N.L., December 2010
The members of the thesis committee hereby recommend the thesis of Juan Paulo de Anda González to be accepted as partial fulfillment of the requirements for the Degree of Master of Science, in:

Information Technology

Thesis Committee:

Guillermo Jiménez Pérez, Ph. D.
Thesis advisor

Miguel Angel Pérez Guardado, Ph. D.
Committee Member, Reviewer

Alejandro Parra Briones, M. Sc.
Committee Member, Reviewer

Ramón Felipe Brena Pinero, Ph. D.
Director of the Graduate Programs in Intelligent Systems and Information Technologies
December 2010
Virtual Organization Process Integration and Execution Through Grid Computing.

BY:
Juan Paulo de Anda González

THESIS

Presented to the Graduate Program in Mechatronics and Information Technologies

This Thesis is a partial requirement for the degree of Master of Science in:

Information Technology

Instituto Tecnológico y de Estudios Superiores de Monterrey
Monterrey Campus

December 2010
To God, for his mercifulness and countless blessings.

To my parents, Armando and Layda, for their infinite love.

To my uncle William as well as my Aunt Guillermina, for having faith in me.

To my brother Armando and his wife Paola for their support and guidance.

To my sisters Layda and Diana, for pushing my back till the end.

To my many relatives and friends who showed neverending support in this project.
Acknowledgements

First of all, I want to thank the members of my dissertation committee. Thanks to my principal advisor, Dr. Guillermo Jiménez Pérez, for being an exemplary mentor.

All my endeavors could not have been realized without the help of my relatives: my uncle Lic. Miguel de Anda and my aunt Sonia, my aunt Sayde who’s my second mother, my godparents Ruben and Martha, as well as everyone else in the family for their support.

Finally I must thank all the professors and fellow students during my stance in these years of graduate and undergraduate school at UAC and ITESM. Because of them I learned more than engineering and much more than what institutions can teach.
Abstract

We find ourselves in times where globalization presses the enterprises to overcome new challenges to be able to remain competitive in the free market. The Small and Medium Enterprises (SMEs) are faced with more difficult challenges to be able to compete with bigger enterprises given that they don’t have a great production capacity. A solution for the SMEs is to create new forms of organization. Enterprises are starting to organize themselves into networks of collaboration (Collaborative Networked Organizations, CNOs) to increase their abilities and capacities to compete. Collaborative Networked Organizations enable the creation of Virtual Organizations (VOs.)

With the creation of VOs, the small enterprises increase their potential to obtain bigger contracts and increment their profit. An obstacle that VOs are faced with, is that the enterprises that comprise them are geographically dispersed; a problem which can result in costly expenses for the members of the VOs and is not always possible to achieve. An initiative is required in order to create tools that support the integration of the VOs to business processes that contain all activities in the life cycle. The creation of various tools that would help organize and administer the communication of the VOs’ members would be an important contribution to the problem but can be a challenge to implement.

This dissertation applies Grid Computing as a viable technology to solve this dilemma. Grid Computing is a scientific computational model to solve complex problems through a parallel infrastructure. This technology was originally intended to be used to solve computational problems and the model began to be used in a higher level, that is, at corporate level. The creation of a similar framework has already been approached by using Web Services; this thesis demonstrates that by utilizing Grid Computing, an alternate tool for collaborative process can be discovered.
# Contents

1 Introduction 1
   1.1 Background ................................................. 1
   1.2 Problem Definition ......................................... 4
   1.3 Objectives ................................................. 6
   1.4 Hypothesis ................................................. 6
   1.5 Document Organization ...................................... 6

2 Collaborative Networks 9
   2.1 Value Networks ............................................ 9
   2.2 Collaborative Network Organizations ......................... 10
     2.2.1 ECOLEAD Project ....................................... 11
   2.3 Virtual Organizations ...................................... 13
     2.3.1 Why VOs? ............................................. 14
     2.3.2 Virtual Organization Development ....................... 15
   2.4 ICT infrastructure ......................................... 16
     2.4.1 ICT-I Architecture .................................. 17
     2.4.2 ICT-I Functions ....................................... 17
     2.4.3 Related Projects ...................................... 19
     2.4.4 Summary ............................................. 21

3 Business Process Management 23
   3.1 BPM ......................................................... 23
   3.2 The Essential Elements of a BPM System ..................... 24
   3.3 BPM Justification and Focus ................................ 25
   3.4 Summary ................................................. 25

4 Service-Oriented Architecture 27
   4.1 What is a service? ......................................... 27
   4.2 What is SOA? ............................................. 27
   4.3 Web Services ............................................. 28
   4.4 Benefits of SOA ........................................... 29
   4.5 Adoption strategies of SOA ................................ 30
   4.6 SOA applications in the business world .................... 31
     4.6.1 Sending automation of client’s requests. .............. 32
# Conclusion and Future Work

8.1 Wrap-up ................................................................. 67  
8.2 Future Work ............................................................ 68 
8.3 Summary ................................................................. 69  

## Appendices

A Submit Files ............................................................ 73  
B Dag Files ................................................................. 77  
C Source Code ............................................................. 89  
D Out Files ................................................................. 93  
E Log Files ................................................................. 101
Chapter 1

Introduction

This thesis presents grid computing as an alternative to integrate business processes with various enterprises working collaboratively in a business opportunity. The main subjects involved in this research are Collaborative Organizations, Business Process Management, and Grid Computing.

This research is located in the context of collaboration networks, as stated by the ECOLEAD, (European Collaborative Networked Organisations Leadership Initiative). This project was established to create networks of collaboration between Small and Medium size Enterprises (SME) in order to compete in a high demand business that only large companies can handle [8].

This chapter will give an overview of the research. The chapter starts with the background, describes the project’s context, then continues with the definition of the problem, a justification, an definition of goals and proposes a hypothesis. Finally, the organization of the document is presented.

1.1 Background

The possibility of rapidly finding a set of partners that best fit a business opportunity and quickly configure them into a collaborative network to exploit that opportunity seems indeed a desirable scenario to face the challenges of market turbulence. The same idea is also very appealing in other non-business oriented contexts. An extreme case being the incident management and disaster rescuing processes, when it is necessary to very rapidly engage and coordinate activities of a large number of entities (for example: fire brigades, police, hospitals, local government, non-governmental organizations). This very idea of groups of organizations being able to rapidly configure themselves into some form of mission/goal-oriented collaborative form embeds the notion of great agility[10].

A Collaborative Networked Organization (CNO) is created by SMEs to establish a relationship between enterprises with related lines of business, common business practices as well as similar infrastructures and standards. With an organization such as this one, Virtual Organizations (VOs) can be created dynamically when a business
opportunity arises, something which single enterprises could not possibly take advantage of alone [11]. A Virtual Organization is a temporal alliance of legally independent business entities (SMEs) that are united to share their common abilities or markets to better respond to the business opportunities, and such cooperation is supported by computer networks [11].

A form of Collaborative Network Organization is a Virtual Organization Breeding Environment (VBE). A VBE is a long standing business relationship between enterprises driven by the willingness of enterprises to cooperate and anchored on common business practices, supporting institutions, and common infrastructures and ontologies. A VBE forms a business environment where collaborative experience is built and where dynamic Virtual Organizations (VO), can be created whenever business opportunities arise [8].

A VBE is managed by a Virtual Enterprise Broker (VEB). A VEB is a business entity responsible for searching opportunities in the global environment and enables the creation of Virtual Organizations. The Virtual Enterprise Broker performs the processes of partner search and partner selection, and configures suitable infrastructures for VO formation, commitment (physical, legal, social/cultural, information) and execution [40].

The life cycle of a VO is typically restricted: it is created from the network for a business opportunity and dissolved after the task has been completed [8]. A Virtual Enterprise is another form of collaboration similar to a VO without the “legally independent” restriction on the organizations. To avoid confusion, this research will focus on the term Virtual Organizations instead of Virtual Enterprises.

As illustrated in figure 1.1, the relationship between a VBE and a VO can be seen and how each one has independent creation processes. A VBE is created as a long term “controlled border” association and its members are recruited from the “open universe” of organizations according to the criteria defined by the VBE creator or administrators. A VO is a temporary organization triggered by a specific business/collaboration opportunity [40].

In simple terms, the life cycle of VO can be summarized into three stages. The first stage consists of the VO formation according to the nature of the business opportunity. This is done by grouping the enterprises which are more apt for the business process in question. The following stage is the creation of an adequate infrastructure for the management of the operation and collaboration among the enterprises that compose it. Once the business opportunity is finished, the final stage consists in dissolving the VO and evaluating it as an individual entity (like an individual evaluation of each of its members to use this information for future VOs [8]). There are various functional requirements found in each one of the stages and some, depending on the nature of the business opportunity can be challenging to the Virtual Enterprise Broker. In order to point out these requirements and how this dissertation handles them, a deeper understating of the VO creation will follow.

The Virtual Organization creation process can be described as Figure 1.2 shows (according to [11]).
Figure 1.1: Virtual Organization Breeding Environment and Virtual Organization Creation (adapted from [8])

Figure 1.2: Virtual Organization Creation (adapted from [11])

This process will be described in detail on the following chapters, the following is a brief list of the Virtual Creation steps[11]:

2. Rough VO planning.
3. Partners search and selection.
4. Negotiation.
5. Contracting.

6. VO launching.

According to [11], all steps involving creation of a VO are supported by specific Business Process Management Models. These models are created by the Broker to manage the workflow of the business processes.

The processes of VO creation and operation fall under a certain infrastructure. ECOLEAD proposes an Information and Communication Technology (ICT) infrastructure for Collaborative Networked Organizations (CNO) doing businesses together. This infrastructure should support interoperability along the operation of the different CNO scenarios, starting from business and services interoperability support among those entities involved in a CNO to more concrete interoperability support for accessing legacy applications; and considering security aspects at different levels.

This dissertation shows how Grid computing can be applied to both handle and integrate some functionalities required in the ICT infrastructure. The methods and technologies that will be implemented are the Business Process Management (which will function as the mediator of collaboration among the enterprises) and the Grid Computing, with the purpose of creating a supporting infrastructure for the broker to establish the formation and execution between processes and the enterprise’s applications within the members of the VO (as seen in figure 1.3). A previous approach to integration was done using Web Services, this will be detailed later in the document.

The process of a collaborative organization unites different enterprises that act as “resources” for a common purpose. Grid Computing administers the flow among “resources” during the operation of a complex process. The enterprises can be viewed as resources used in Grid Computing, since it demonstrates to have a similar methodology to the administration of the businesses’ process in a collaborative environment. If Grid Computing helps overcome various issues required during the operation and execution VO collaborative process, it can prove to be a viable tool for this sort of scenarios. The issues will be described later in the document.

1.2 Problem Definition

Inside a VBE the SMEs are willing to collaborate by forming VOs to be able to approach a large business opportunity [43]. The entities of these VOs require an infrastructure of inter-operation, rules of operation and common agreements of cooperation, among other things, so that the execution of the collaboration can be achieved faster and have greater probability of success. Each business opportunity is to follow several business processes. The business process will normally include some members of a CNO, but it must be defined in such a way that many members are able to participate; as stated before, a step-by-step strategy where different enterprises can become part of each step. Creating the template of the process and all the activities to follow and all the different operations that the members will perform can be described. A Virtual Enterprise Broker must designate the steps to perform to all the members of the VOs. These
templates must be utilized by the members to be able to obtain and send information to the templates (with their software applications). Grid Computing will be used to establish the standards of communication and integration to the business processes.

Grid Computing is a hardware and software infrastructure that provides dependable, consistent, pervasive and inexpensive access to high-end computational capabilities. Its goal is to produce a giant virtual entity composed of a collection of heterogeneous systems that share their resources [6]. Originally intended for scientific purposes, Grid Computing is proposed as an infrastructure for the formation and execution of virtual organization processes.

Since Virtual Organizations are composed of various independent SMEs, it can be assumed that the location of these enterprises can be far from each other. The emergence of a strong communication functionality is vital for the Virtual Organization’s creation and operation. The practical implantation of the VO paradigm is far from its initial expectations[9]. This investigation will determine to what degree the Grid Computing Infrastructure can help the creation and operation of a Virtual Organization.
1.3 Objectives

This investigation analyzes to what extent the Grid Computing framework can facilitate the VO creation and operation. This includes the integration of a business process to the VO members’ activities with the use of the Grid infrastructure. To achieve this, the following specific objectives are defined:

- Investigate Virtual Organization Scenarios.
- Choose a Grid Computing Technology that can be applied to define and execute the Virtual Organization Process.
- Investigate which functionalities can or can not be handled by Grid Computing.
- Create a prototype to integrate the Virtual Organization Processes to the Grid Infrastructure.
- Analyze and present results of the prototype based on Grid Infrastructure

1.4 Hypothesis

Grid Computing enables the definition and execution of Collaborative Business Processes within Virtual Organizations.

1.5 Document Organization

This dissertation is organized in nine chapters covering all important aspects of the research.

This chapter presented the main concepts used in the document, defines the problem to solve, justifies the project, exposes the goals to accomplish and finally proposes a hypothesis.

The second chapter introduces the reader to the Collaborative Networks topic according to the ECOLEAD project context. It also describes the fundamental elements involved in the project, which are VBE and VO. These concepts are the basis of this project since the problem is born from the VO’s need for collaboration. It continues to describe the process of VO formation and execution.

The third chapter exposes Business Process Management and how it supports the definition of the VO processes for their definition and execution.

The fourth chapter continues with Software Oriented Architecture and helps understand how the Collaborative Process is Service oriented.

The fifth chapter describes Grid Computing and how it will be used in this research.

The sixth chapter focuses on the chosen Grid Framework: CONDOR. It describes the technical aspects needed to configure the framework for this research.
The seventh chapter describes the case study in which the research was conducted.

The eight chapter describes the implementation of a grid infrastructure to integrate the collaborative processes described in the case study. It also exposes the results and findings of the research.

The ninth and final chapter concludes the research, summarizing the objectives achieved and invites further research.
Chapter 2

Collaborative Networks

This chapter presents the concepts necessary to have a clear understanding of this dissertation. The chapter focuses into the context in which this dissertation revolves around: Virtual Organizations for Collaboration Projects and the Information and Communication Technologies (ICT) infrastructure that supports them. The information is based on the research done by the European Collaborative Networked Organizations LEADership initiative (ECOLEAD).

2.1 Value Networks

A value network is any web of relationships that generates tangible and intangible value through complex dynamic exchanges between two or more individuals, groups, or organizations. Any such web (i.e., organization, group of organizations, or purposeful network), in which participants are engaged in tangible and/or intangible exchanges, can be viewed as a value network, whether it is in private industry, government, or the public sector. All exchanges of goods, services or revenue, including all transactions involving contracts, invoices, return receipt of orders, request for proposals, confirmations and payments are considered to be of tangible value. Knowledge and benefits are considered of intangible value (for example: strategic information, planning knowledge, process knowledge, technical know-how, collaborative design and policy development)[3]. A simple example of a Value Network is a customer (this is a role) goes to a store and buys groceries from the cashier (another role). Money is paid in return for goods: vegetables. If the customer lives in a small town and has an ongoing patronage relationship with the cashier, there might be an intangible value exchange of information about their families and the neighborhood.

This research will touch upon a form of Value Network proposed by the ECOLEAD Project: Collaborative Network Organization.
2.2 Collaborative Network Organizations

The emerging paradigm of Collaborative networked organizations (CNOs), or simply collaborative networks, fundamentally changes the way that commercial, industrial, cultural, and social activities are organized. In addition to the rapid evolution of traditional supply chain and outsourcing practices, a growing trend nowadays consists of tasks performed by autonomous teams of a small number of people or small and medium enterprises (SMEs), set up as independent contractors or small firms and linked by a network. These teams usually come together, sometimes in temporary arrangements, to tackle various projects, and may dissolve once the work is done [9].

It is frequently stated that collaborative networks provide a basis for competitiveness, world/excellence, and agility in turbulent market conditions. These networks can support SMEs to identify and exploit new business potentials, to boost innovation, and to increase their knowledge. The networking of SMEs with large scale enterprises also contributes to the success of both kinds of companies in the global market. Nevertheless the practical implantation of this paradigm is still far from the initial expectations[9].

Networks represent the different forms of collaboration among groups of autonomous entities structured such as [9]:

- **Virtual enterprise.** Temporary alliance of enterprises that come together to share skills or core competencies and resources in order to better respond to business opportunities, and whose cooperation is supported by computer networks.

- **Extended enterprise.** Organization in which a dominant enterprise “extends” its boundaries to all or some of its suppliers.

- **Virtual Organization (VO).** Similar to a virtual enterprise, comprising a set of (legally) independent enterprises that share resources and skills to achieve its goal, but not only limited to an alliance of for profit enterprises.

- **Dynamic VO.** It typically refers to a VO that is established in a short time to respond to a competitive market opportunity, and has a short life cycle, dissolving when the short-term purpose of the VO is accomplished.

- **VO Breeding Environment (VBE).** It represents an association or pool of organizations and their related supporting institutions that have both the potential and the will to cooperate with each other through the establishment of a “base” long-term cooperation agreement.

- **Professional virtual community (PVC).** It provides the necessary environments for professionals to develop their skills and to share the knowledge and experience of their professions using emerging communication technology as their cooperation platform, therefore reducing constraints of geographic, organizational, or temporal boundaries [33].
To understand both CNOs and its benefits it is necessary to understand a little more about the ECOLEAD project and its goals. The following section provides more details of the ECOLEAD project.

2.2.1 ECOLEAD Project

The European Collaborative networked Organizations LEADership initiative (ECOLEAD) was a 4 year project started in April 2004, involving 20 partners from 14 countries across Europe and America. ECOLEAD aimed to create strong foundations and mechanisms needed to establish the most advanced collaborative and network-based industry society. It’s vision was that in a ten years most enterprises will be part of some sustainable collaborative networks that would have the tools and knowledge necessary to form dynamic virtual organizations in response to fast changing market conditions.

The objective of ECOLEAD was to create strong foundations and mechanisms needed to establish the most advanced collaborative and network-based industry society in Europe: In ten years most enterprises would have the tools and knowledge necessary to create dynamic virtual organizations within their established collaborative networks and breeding environments.

The fundamental assumption in ECOLEAD is that a substantial impact in materialising networked collaborative business ecosystems requires a holistic approach. Area’s complexity and the multiple inter-dependencies among the involved business entities, social actors, and technologies, substantial breakthroughs cannot be achieved with incremental innovation in isolated areas. On the other hand, project plans must remain manageable. Thus ECOLEAD’s Architecture addresses the most fundamental and inter-related focus areas, which form the basis for dynamic and sustainable networked organizations: the VO Breeding Environments and Dynamic Virtual Organizations. In addition to these vertical focus areas, the holistic approach is reinforced and sustained on two horizontal areas: the theoretical foundation for collaborative networks and a horizontal Information and Communication Technologies (ICT) infrastructure. The horizontal activities support and affect all three vertical focus areas. The theoretical foundation shall provide the basis for technology-independent understanding of the area and its phenomena. The existence of an invisible, low-cost ICT infrastructure is a pre-condition for the establishment of truly dynamic collaborative networks. ECOLEAD was expected to impact industrial competitiveness and societal mechanisms, by providing means to effectively exploit opportunities deriving from the deployment of VOs, and by designing and enabling new professional work paradigms, capable of enacting the knowledge-based society throughout Europe[10]. Figure 2.1 shows the ECOLEAD Architecture mentioned.

ECOLEAD’s approach is to address the modeling and development of service functions to support the full life cycle of VOs, in the context of an underlying breeding environment and in interaction with the professional virtual communities emerging in this environment. 2.2 shows the VO lifecycle, which could be described as follows:
• VO creation. In this phase the set up and configuration of a specific VO is done. ECOLEAD aims the development of a set of generic functions to support the creation and launching of VOs in the context of a VBE. Some isolated functionalities addressed in past projects include: VO planning, partners search and selection, enterprise catalogues, contract negotiation, etc [10].

• VO operation. Devoted to take care of the activities fulfilment, monitoring and coordinating the performance. Coordination of distributed business processes and activities is an important element in the VO operation. ECOLEAD focuses on the development of generic functionalities for the distributed business process management [10].

• VO evolution and dissolution. During the life cycle of a VO it is natural that some partners leave the consortium and be replaced by a new partner. The termination of this collaboration process or even the ending of a VO, are subjects not properly addressed yet. The consequences of the operation of a VO cannot be simply discarded when the VO dissolves. Most of these consequences are of a legal nature and shall be regulated by the cooperation agreements. Environment regulations are also forcing companies to plan provisions regarding the product disposal and recycling after its end of life. Therefore the life cycle of a product might involve a number of VOs. There is also considerable knowledge that can be elicited from the ending cooperation experience, namely the knowledge about what went right, what went wrong, partners performance and reliability, jointly defined business process templates, etc [9].
The collaborative networking paradigm offers an opportunity to develop interoperability of enterprise applications. ECOLEAD assesses the opportunity to make a reality the vision of cross-enterprise co-operative processes and new methods of work relying on the European diversity. A multidisciplinary approach is needed and it must integrate contributions from areas such as: Computer Science, Organization and Social Research, Economic Research, Cognitive Sciences, Operations Research, Process Modelling and Simulation[43].

ECOLEAD also focuses on an ICT infrastructure that can bring “support functionality” that Virtual Breeding Environments require for a suitable creation and operation of a Virtual Organization. This functionality includes[66]:

- Maintenance and Management of well classified repositories / catalogs of information regarding the profile, competencies, resources, products, services, etc. about its member organizations. This helps the Broker or VO coordinator to replace a failing partner in the VO with another of the same capabilities.

- The establishment of a credibility record of enterprises based on their past performance, etc. This helps the creation of future VOs.

- The establishment of effective mechanisms to locate best match enterprise for the required skill/competency for the involvement in the VO. This reduces the cost/time necessary to search/find suitable partners (based on competency, knowledge, services, etc.).

- Fast contract negotiation steps for establishment of the VOs.

- Dynamic reconfiguration of the VOs, which reduces the risk of big losses due to partner failures.

- Establishment of commonly accepted infrastructure business culture to be adopted by interregional VOs, to include the following among others:
  - A common base ICT infrastructure (necessary for VO collaboration), to reduce the set up time during the VO formation.
  - Introduction of common cooperative business rules, that can be measured by common metrics for evaluation of member’s credibility and performance.
  - Provision of the base common ontology for specialization of both the generic and the sector dependent competency / skills / capacity, etc.
  - Offering access to a “bag of assets”, containing common information of interest as well as a set of basic software tools

2.3 Virtual Organizations

A Virtual Organization (VO) is composed of legally independent enterprises that contribute their own particular abilities to the same cause. To accomplish this, Virtual
Industry Clusters (VIC) (a practical form of adding companies from diverse industries to have a wider variety of markets and opportunities) and a Virtual Enterprise Broker (VEB) (business entity who functions as an agent for the selection of partners among the different enterprises) are used[11]. Nowadays it is imperative to form VOs quickly due to the business opportunities that happen to arise both unexpectedly and frequently.

2.3.1 Why VOs?

Rapidly finding partners for a business opportunity is the first important challenge of a collaborative project. That search comes with many obstacles such as the time constraint, the pricy cost of carrying out the search, and lack of preparation of several organizations, among others. There have been attempts to tackle the previously mentioned problems in order to create a more efficient collaborative network. An example of one of those attempts is the Manual Assisted approach. Manual Assisted approach, in Human Resource management, was initially determined by the acknowledged inability of traditional psychological and cultural tests in predicting successful job performances. A typical difficulty is the access to a reliable profile (the set of required skills of a potential partner) and performance information as well as the time spent with the preliminary filtering of potential candidates[11].

Another attempt was the Multi-agent based approach. A multi-agent systems(MAS) application assumes a virtual market place where enterprises, represented by agents that are geographically distributed and possibly not known in advance, can meet each other and cooperate in order to achieve a common business goal. Also, a MAS architecture is proposed to model the electronic market to support the formation of the VO. In addition to the agents representing the enterprises, there is a market agent (coordinator or broker) that is created and inserted in the MAS community when a business opportunity is found. But the attempt to reach a fully automated decision-making process, although an interesting academic exercise, is quite unrealistic in this application domain[11].

A different attempt was the Service federation based approach. The service federation based approach (also known as “service market,” “service promoter node,” or “service portal”) assumes the existence of one entity that keeps a catalog of services where service provider companies (potential members) publish their service offers. This approach has many problems among them: that services are not always available; there is no specific level of awareness of the service provider, and perhaps the most important, that most example developments so far are for ecommerce, not for collaborative activities[39].

With the use of a VO, past challenges can be achieved more efficiently. VOs can give a number of benefits by adjusting themselves according to the needs and opportunities of the market, as well as remaining operational as long as these opportunities persist. Other benefits include the fast response to opportunities or needs and introducing new business opportunities as well as new business practices for the organizations involved.[39] The following sections describe VOs
2.3.2 Virtual Organization Development

There are several steps to take to develop a VO, the steps are listed as follows:

1. Collaboration Opportunity Characterization: this step involves the identification and characterization of a new Collaboration Opportunity that will trigger the formation of a new VO. A collaboration opportunity might be external, originated by a (potential) customer and detected by a VBE member acting as a broker.

2. Rough VO planning: determination of a rough structure of the potential VO, identifying the required competencies and capacities, as well as the organizational form of the VO and corresponding roles.

3. Partners search and selection: this step is devoted to the identification of potential partners within the Collaborative Organization, and their assessment and selection. Issues to consider: Elements for search and selection (technical, economical, reliability indicators, preferences); matching algorithms; (multi-criteria) selection criteria; optimization; assessment (preparedness, etc); consideration of collaboration history / record; external search (if the internal offer is insufficient); etc.

4. Negotiation: is an iterative process to reach agreements and align needs with offers. It can be seen as complementary to the partners' selection process and might in fact require going back to the previous step(s) if a solution cannot be found with the current configuration of partners.

5. Detailed VO planning: once partners have been selected and collaboration agreements are reached, this step addresses the refinement of the VO plan and its governance principles.

6. Contracting: involves the formulation and modeling of contracts and agreements as well as the contracting process itself, before the VO can effectively be launched. A contract is an agreement between two or more competent parties in which an offer is made and accepted, and each party benefits. A contract defines the duties, rights and obligations of the parties, remedy clauses as well as other clauses that are important to characterize the goal of the contract. An agreement is an arrangement between parties regarding a method of action. The goal of this arrangement is to regulate the cooperation actions among partners, and it is always associated to a contract.

7. VO launching: the last phase of the VO creation process, i.e. putting the VO into operation, is responsible for tasks such as configuration of the ICT infrastructure, instantiation and orchestration of the collaboration spaces, assignment and set up of resources / activation of services, notification to the involved members, and manifestation of the new VO in the VBE.
There are several functionalities needed for the creation of a Virtual organization just mentioned. The table 2.1 describes the tools and functionalities required for each one of the steps:

<table>
<thead>
<tr>
<th>Step</th>
<th>Type of Functionality</th>
<th>Functionalities required</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Collaboration Opportunity Charac-</td>
<td>Preparatory Specifica-</td>
<td>CO feasibility analysis</td>
</tr>
<tr>
<td>terization</td>
<td>tion</td>
<td>Rough VO architecture design (structure, topology, main roles); Identification of needed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>competencies and capacities; Rough collaboration process modeling; and Simulation for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>preliminary assessment</td>
</tr>
<tr>
<td>2. Rough VO planning</td>
<td>Preparatory Specifica-</td>
<td>Partners search and selection; Assessment of partners preparedness</td>
</tr>
<tr>
<td></td>
<td>tion</td>
<td>Specification of preferences; Negotiation protocols and methods</td>
</tr>
<tr>
<td>3. Partners search and selection</td>
<td>Consortium Formation</td>
<td>Mechanisms to assign roles and responsibilities; Definition of operating rules for the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VO</td>
</tr>
<tr>
<td>4. Negotiation</td>
<td>Consortium Formation</td>
<td>Contract modeling; Contract process support</td>
</tr>
<tr>
<td>5. Detailed VO planning</td>
<td>Support</td>
<td>Configuration and setup of the infrastructure and resources; Manifestation of the VO;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Notification to the partners</td>
</tr>
<tr>
<td>6. Contracting</td>
<td>Support</td>
<td></td>
</tr>
<tr>
<td>7. VO launching</td>
<td>Finalization</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.1: Tools Required for the Creation of VOs (adapted from [11])

### 2.4 ICT Infrastructure

As mentioned before, in order for Collaborative Networks to create and operate VOs, the use of a simple and affordable ICT infrastructure is vital.

A well designed ICT Infrastructure (ICT-I) should provide the support services that are involved and found in the VO creation process and the enterprises interoperation. It is the base enabler for safe and coordinated interactions among the VO members. In other words, it plays the role of a VO “operating system” or executor,
hiding the details of the collaborative network “machinery” from the users and application programs. Support services consist of a set of vertical application tools that will be plugged into the horizontal ICT Infrastructure [8].

Many interoperability issues arise in this infrastructure, because of this, a solution that is fast and simple is needed, in this case, the ECOLEAD Infrastructure approach chosen was Service Oriented Architecture (SOA).

2.4.1 ICT-I Architecture

According to [43] an ICT-I Architecture in the ECOLEAD context needs to solve the following interoperability issues:

1. Access from multiple Devices: ECOLEAD must support CNO actors accessing ECOLEAD services through diverse devices. Translated to technical requirements this means support for devices interoperability as the capability to offer to any user, the connectivity to any device, on any network, with any content. Each of the CNOs have their respective portals through which users can have access to (all of some type of) vertical services. Possible external providers may have portals as well.

2. Support for Business Collaboration: ECOLEAD ICT-I architecture must be designed to support truly integration of businesses from different organizations collaborating in a CNO as PVC, VBEs and VOs, as the interoperability among services from different organizations collaborating in a CNO. The analysis of ICT-I support for business collaboration has been performed with the focus on the main requirement to support business process definition as a set of services.

3. Seamless Integration of Services and Applications: ECOLEAD’s ICT-I must facilitate and support dynamic integration of services and applications, which follows from one of the major requirements imposed to the ICT-I platform to support collaboration scenarios in a changing environment where new organizations appear to offer and demand services.

4. Heterogeneous Applications Integration: Usually, services and operations defined for different CNO environments require connection with existing applications and/or data bases running at the organizations composing the CNO.

5. Security interoperability: The effective application of ICT-I architecture must support the operation of different CNO actors, which are demanding quality of services ensuring the integrity, confidentiality and security aspects through their interactions within the CNO platform.

2.4.2 ICT-I Functions

The ICT-I functioning comprises 3 main steps: processes design/modeling, the orchestration of the services involved in these processes, and their execution [5].
Process Design/Modeling

All processes must be previously designed and modeled for each business case. The process designing could be done with Business Process Management diagrams or assisted by some auxiliary tool (maybe also a service, with a front-end application). During Modeling, every service within the task needs to be described uniquely, usually indicating characteristics of it or specific information about it.

Process Orchestration and Execution

Process orchestration is performed by an execution engine, which is responsible for identifying the set of services required for a previously defined process, then dynamically find, select orchestrate and further execute (coordinate) them. As this is a Service Oriented Application approach, only references to services are used. The physical localization of services does not matter. Upcoming chapters will talk about how grid computing fits well for this purpose.

Not every time a one-to-one relation can be traced between business tasks (in a process) and services (in an orchestration). In some cases several services will be needed to perform a single task. The rough idea of this approach is the following: There is ontology support when the services of the task at hand are defined by the CNO Broker. This ontology is taken by an “intelligent” orchestration service (a service provided by the ICT-I) that will “deduce” dynamically the services (and their sequence) that should be invoked to accomplish the given process.

The main requirements of process orchestration and execution are described as follows[5]:

- **Dynamics.** As the services execution engine will only invoke the required services once a given process is necessary to be executed, all steps involved to the invocations may be dynamically taken.

- **Finding.** In order to carry out the process execution, the services need to be found. As the search is performed, a set of services, corresponding to the search parameters, is obtained. Depending on how the services were specified on the choreography plan (specifically, or in a more general way), a broader (or not) range of services will be returned. The most suitable service of these will be chosen in the selecting phase.

- **Selection.** According to the metadata of the services returned by the finding phase and the criteria applied, the most suitable service is selected. These criteria comprise security, ontological, etc. parameters.

- **Execution.** Once the services are selected they are, in a coordinated way, executed conforming their orchestration.

The services can be modeled as independent/separated services or can be grouped and modeled as methods/operations of the execution engine service. Besides that, the
finding service can gather all services prior to their execution or can execute each one as soon as it is found [5].

To get a better understanding of VO creation and operation one can think of an assembly project. The items to be assembled are office chairs but the quantity is large enough that many enterprises within the CNO have to become involved for the various processes within the task (Gathering the material, building the separate pieces, assembling them all together, etc). It is clear that a central entity needs to manage all enterprises to work together for the big task. With an ICT Infrastructure the management can be achieved by one or more players within the VO. This can be achieved due to the ICT-I taking care of the communication and coordination between the enterprises making the management easier.

Grid Computing will be used to approach the issues described in this section of process orchestration and execution. Given an intended orchestration defined by a business process, the grid can dynamically select those services offered by the enterprises that best fit to process and then integrate them. The following section describes other projects that have used different approaches to solve these issues.

2.4.3 Related Projects

The application integration problem has been approached in various ways. This section will present several research works that have been successful in related material to this dissertation.

“Web services based supporting platform for virtual enterprise” (a project developed by [66]) has five Web Services layers: Data Access, Fundamental, Common, Integration and Virtual Enterprise. All five layers resolve several problems found, however, the Integration layer is going to be the focus in this project because it’s more applicable in this case. The Integration layer is formed in such a way that it shares data and resources companies supply. Workflow management must be used in the integration service, the workflow itself could be simple or extremely complex depending on the internal or external web services contained on the main workflow.

An interesting proposition by Yang et al [31] is that of an e-service based architecture used for cooperative processes (VO business processes) and information transferring from one participating organization to another. The organizations must follow several steps to build VO applications based on e-services:

- **Service Presentation**: The first step offers business application functionality in the form of e-services for online access. Typical technologies used at this step are XML, Java applets and servlets, CGI and home grown software for the communication with the back-end databases.

- **Service Publication**: The second step involves the publication of the service interfaces and semantics in a form that can be understood and queried by other organizations when developing a VO application. They have developed a service description language (SDL) which is similar to WSDL.
- **Service Selection:** Tools and facilities should be offered for querying the service descriptions with the aim of understanding their functionality and appropriateness for an application. At this step they are proposing an interesting way to dynamically set up the business logic for a VO.

- **Service Composition:** The final step is the availability of facilities that enable the construction of VO application by means of combining existing services and composing new services on the basis a service library. This step involves issues such as service composability, compatibility, conformance and substitutability.

On December 2002 the Web Services Outsourcing Manager Framework was developed by IBM Watson Research [65]. The way it was developed was through a mathematical model for dynamic business processes configuration that used available Web Services that were needed by the customers. The XML search tool used by Web Services on a requirements document is used to produce the final business process. This process also can be generated on a Web service execution language such as BPEL, WSFL or XLANG.

A web Services-based Business Interactions Manager was created in 2005 [64] to support electronic commerce applications, and it covers the following tasks: (1) abstracting, specifying (in terms of activities), and categorizing the business interactions into enterprise, customers, suppliers, and partners’ interactions, (2) insulating the core business activities from the interaction activities to make these two types of activities independent, (3) dedicating to the interaction activities a specific artifact called Business Interactions Manager (BIM), (3) specifying the functionality of the BIM. This specification aims at overcoming the limitations of the current technologies, especially the de facto connecting technology, that is Web services, in order to specify a Web services-based BIM as a backbone of Web Services-Oriented Architecture (WSOA), and (4) implementing a specific instance of the Web services-based BIM for each category of e-commerce with regard to the business specifics.

A simulation-based approach was developed by Tewoldeberhan [61] to improve orchestration of supply chain business processes. The simulation-based approach uses the BPM standard BPEL4WS. It also presents a fictitious end-to-end supply chain case study to exhibit and evaluate the approach.

In 2006 Manuel Ocampo[43] developed a Business Process Integration toolkit with the use of SOA / Web Services. The research focused on the use of XML as the language to define the processes and Web services as the connection driver between a business process and enterprises’ applications. He developed a software kit integrate business processes as plug and play components to the enterprises. Ocampo’s research is focused on the first steps of creating a VO (Collaboration Opportunity Characterization and VO Planning), while this research is focused on the last steps(execution and orchestration). This dissertation also uses the ECOLEAD definition and standards for Virtual Organizations, the basis of both research are the same, but the technology used as well as the VO steps focused on are different.
2.4.4 Summary

This chapter described the context in which this dissertation focuses. CNO and VO were described as well as their creation and operation. Then the ICT-I infrastructure was described as well as the interoperability necessities that arise during the orchestration and execution of the process. The following chapters describe the technologies used to model and design the services, as well as the technology used for process orchestration and execution.
Chapter 3

Business Process Management

The previous chapter explained why a business process approach is necessary to describe the collaboration of enterprises in a VO. This chapter presents methodologies that are used to define the collaborative processes found in a VO. It describes what Business Process Management is and continues with its purpose related to this dissertation.

3.1 BPM

Business process management (BPM) is concerned with managing change to improve business processes by treating them as assets to be valued. It is a structured approach employing methods, policies, metrics, management practices and software tools to manage and continuously optimize an organization’s activities and processes. BPM unifies the disciplines of Process Modeling, Simulation, Workflow, Enterprise Application Integration (EAI), and Business-to-Business (B2B) integration into a single standard. The fact that Business Process Management is a new initiative might lead us to believe that business processes have not been considered previously. This is of course not true, many organizations have modeled and managed their business processes for years, using a diverse mixture of tools and techniques, such as Six Sigma, Total Quality Management, Business Process reengineering, to name a few. Some of these techniques have been partially successful, while others have failed, BPM has been chosen because of the standards it offers as well as the control of the complete lifecycle. BPM also offers guidance for the design and execution of business processes.[29][41][63].

A business process is a collection of related, structured activities that produce a service or product that meets the needs of a client. These processes are critical to any organization as they generate revenue and often represent a significant proportion of costs. Each business process contributes to one or more business goals. To gain information on how efficient the business processes are actually conducted and whether the business goals are attained by the business processes, controlling activities are conducted. Key Performance Indicators (KPI) of business processes are determined. KPIs are financial and non-financial metrics used to help an organization define and measure progress toward organizational goals. KPIs can be delivered through Business Intelli-
gence techniques to assess the present state of the business and to assist in prescribing a course of action. [47]

Managing process change cannot be an ad-hoc process, it requires management to exercise control over the discovery, architecture, design, and deployment of processes. For management to understand the architecture, design, and deployment of processes, we need business modeling and business execution language standards.

Business process management is the general term for the services and tools that support explicit process management (such as process analysis, definition, execution, monitoring and administration), including support for human and application-level interaction. It has proved to be valuable in defining effective business processes of everything from a single department to an entire enterprise and its associated value networks. This function can range from the basic automation of a single manual process (such as accounts payable) to a line of business or an overall business (such as policy underwriting).[32].

3.2 The Essential Elements of a BPM System

A business process management system is a generic software system that is driven by explicit process representations to coordinate the enactment of business processes. Although some of the BPM systems and tools have limited support, many vendors claim to have BPM even if they involve just a part of the functionality required by BPM, despite the low levels of quality that result. From the client’s perspective, it is expected that the prospect leaders have all five elements properly done:[43]:

• Designing tools. This are used to model and define processes, these tools are targeted at business analysts who extract established process flows and design new flows. These flows are then specified in a friendly development environment for future execution.

• A runtime execution engine. This is the underlying state machine that executes the defined process flow. As the process flow is executed, the engine may invoke automated services or tasks that humans have to complete. The services may be provided by applications, legacy or new, or by other enterprises that might be trading partners or outsourcers. The runtime environment maintains the status (state) of each process instance or business event.

• Tools to monitor and manage the flows. Monitoring may cover process performance, degree of completion or out-of-bounds conditions. Process management may cover process termination, compensating processes, load balancing and rerouting.

• Tools for post-completion analysis. These tools use the state data that is archived for business measurement and adjustments.
3.3 BPM Justification and Focus

So why not use a BPM system to solve the VO problem? The reason is there aren’t any BPM systems that focus on Collaboration between multiple entities. According to Samly Kensley[34] the trends toward greater collaboration between various entities have been growing and the current BPM systems lack the proper agility in dynamic orchestration and execution between various entities.

BPM and workflow have been used interchangeably but have a few differences. In general, workflow refers to the mixture of tasks that make up a process. In the past, workflow was also used to refer to a legacy market, thus increasing the confusion. What differs BPM from workflow is the human-to-human flow. System-to-system flow automation is included in BPM among other concepts. This concept is focused on more because it implies the addition of automated participants to help manage a part of a process by reducing the number of human actions involved. Therefore, one key part of a larger set of definitions related to BPM is “human-side,” which relates directly to people and their assignments in fulfilling a goal. Another concept that BPM also focuses on is the “integration side” between systems and the information flow necessary to provide completion of a different kind of process: system interoperability and dynamic data exchange [32].

This dissertation addresses this last point as exposed by Sinur [32], “System interoperability and dynamic data exchange” as both are issues found in the ICT Architecture required for the VO. As mentioned before, BPM implies process modeling, EAI, and B2B integration. One of the objectives of this dissertation is to aproach this issues with Grid Computing, it will be described in upcoming chapters.

3.4 Summary

This chapter has described what BPM is and the way it can help define and manage the flow of business processes. The next chapter will describe how SOA can be used to address process modeling, EAI, and B2B integration. The exposure of the essential elements of a BPM system was made and will be touched upon on again through a Grid Computing approach.
Chapter 4

Service-Oriented Architecture

The goal of this dissertation is to demonstrate to what extent Grid Computing can integrate the service applications of each individual business process with the corresponding members of the VO. An understanding of Service Applications is required. This chapter will describe the Service Oriented Architecture (SOA) and the benefits it offers at corporate and IT levels. It will also describe the technologies behind SOA and how they address enterprise integration. It is this approach that will be used by the Grid Computing implementation.

4.1 What is a service?

A service is a concrete functionality that can be found in the web; it describes its purpose and the way of interacting with it. From the enterprises’ perspective, a service realizes a concrete task: it can correspond to a business process so simple such as introducing or obtaining data like “Client’s Code.” But the services could also be used within a complete application that can provide services of a high level, with a highly complex level, a process that, from the moment it begins to the moment it finishes, can involve various business applications. The services can be of a low-level function or of a high-level and based on this definition; many real differences in performance, maintenance simplicity, and reutilization exist. This process of defining services is normally a conversion realized by objects within a greater reach than that of a structure based on components, in which a the services are defined as a group of reused components, services which can be used to construct new applications or integrate existing programs.

4.2 What is SOA?

SOA is a set of principles for software design and integration where the business applications are divided into individual services that can be utilized apart from the applications which they are part of and of the information platforms that are executed. By being
able to rely on the individual application services like the independent pieces, the enter-
prises will have the possibility of integrating and uniting them in different ways to 
obtain completely new capacities [38]. SOA establishes a blueprint for the integration 
of the independent applications in such a way that its functionalities can be accessed through the web, which would be offered as services. The most common way of implement-
ing this is through Web Services, a technology that is independent from the platform. SOA can divide monolithic applications in a group of services and implement their functionality in a modular form.

SOA takes on a general organization strategy of the IT elements, in order that 
a collection of the distributed systems and complex applications can be transformed into 
a web of integrated resources, simplified and exceptionally flexible. Enterprises need to be able to interconnect the processes, the people and the information with the organization itself, as well as with subsidiaries and commercial associates. The lack of integration between the components of the IT makes it difficult to obtain a fast and effective response to the changes that affect, in a natural way, the businesses. The lack of flexibility generates costs, reduces the capacity of response to the clients, compromises the compliance with the legal norms and negatively affects the productivity of the employees [38].

A well executed SOA project allows to align the IT resources in a more direct 
way with the business objectives, thus winning a greater degree of integration with the clients and providers, proportioning a more precise and accessible business intelligence (in which one could adopt better decisions) and in turn help the enterprises to optimize their internal processes and their flows of information to improve the individual productivity.

4.3 Web Services

The adoption of a solution of a design based on SOA does not demand to implement Web Services. However, Web Services are the most common form of implementing SOA. The Web services are applications that utilize standards for the transport, coding and protocol for exchanging information; support the intercommunication among the systems of any platform and are used in a great variety of places of integration, both within the organizations as well as with the businesses’ associates. Web services are based on a group of standards of communication, like XML are for the representation of data, SOAP (Simple Object Access Protocol) for the exchange of data, and the language of WSDL (Web Services Description Language) to describe the functionalities of a Web Service. More specifications exist, which are generally named as the WS-* architecture, that define different functionalities for the finding of the Web Services, event management, joined archives, security, management and reliability in the exchange of messages and transactions [17].
4.4 Benefits of SOA

Two different levels of SOA benefits for an organization exist: at the corporation-user level and at the IT organization level. The SOA architecture, from the business point of view, in figure 4.1, permits the development of a new generation of dynamic applications that resolve a great quantity of high-level problems, fundamental for the growth and competition. The SOA solutions permit, among other things:

- Improving the decisions. By integrating the access to the services and business information within a group of composed dynamic applications, the management has access to more information and with better quality. By having access to better information in a shorter period of time, the organizations can react faster and in a more agile way when problems or changes emerge.

- Improving the productivity of the employees. An ideal access to the systems and to the information, and the possibility of improving the processes that permit the enterprises to increase the individual productivity of the employees. They can dedicate their energies to the important processes, those that generate an added value, and to collaboration activities, semi structured; instead of accepting the limitations and restrictions imposed by the rigid and monolithic IT systems.

- Boost the relationships with clients and suppliers. If the clients and external providers can count with access to applications and dynamic business services, not only could an advanced collaboration can be permitted, but also the satisfaction of clients and of providers can increase.

Figure 4.1: SOA from the business point of view. [1]
From the point of view of the IT departments (see Figure 4.2), the orientation to services involves a framework in which the creation and maintenance of systems and integrated applications can be simplified along with a formula to alienate the IT resources with the business model and the needs and dynamics in such a way that they are affected. SOA, from the technological point of view, is the result of the constant evolution toward a major decoupling of the layers of an application (presentation, process orchestrations and business services) and gives a greater level of standardization/interoperability of each of these layers.

- More productive and flexible applications. The strategy of orientations to services permits an IT to obtain a greater productivity from the resources of the existing IT and to obtain greater value from them. The orientation to services also permits the development of a new generation of composed applications that offer advanced and multifunctional capacities for the organization with independence of platforms and programming languages that support the base processes.

- Development of faster and cheaper applications. The design of services based on standards facilitates the creation of a reusable service repository that can be combined into services of a greater level and applications composed in response to new necessities of the enterprise. And the use of an environment and a model of unified developments simplifies and homogenizes the creation of applications, from its design and even tests its execution and maintenance.

- More secure and manageable applications. The service-oriented solutions provide a common infrastructure to develop services that are secure, predictable, and manageable. As the business necessities are evolving, SOA facilitates the probability of adding new services and functionalities to manage the critical business processes. The services are accessed, not the applications, and because of that the SOA optimizes the realized inversions in IT boosting the capacity to introduce new and better capacities.

### 4.5 Adoption strategies of SOA

Embarking in a SOA project implies to have to solve a series of challenges, at both organizational levels as well as technical levels. These challenges can become real barriers incapable of overcoming if the idea of SOA being the remedy for all kinds of illnesses is taken. In order for the SOA adoption initiatives to have a satisfactory ending, a series of indispensable conditions must be accomplished[1]:

- Clearly defining the business objectives. The first step at the moment of adopting SOA is to identify with clarity the high-priority enterprise problems or challenges.

- Clearly defining the reach of the SOA project. The objective of any SOA project must not consist of renovating in an indiscriminate and massive manner all the
Impact of SOA in the evolution of the technologies of the information from the point of view of development of applications.

Figure 4.2: SOA from the technology point of view. [1]

IT infrastructure. The real objective of each SOA initiative must be to respond to the concrete business needs and to create solutions in discrete, increasing, and repetitive steps.

- Avoid introducing SOA without real motives that would justify it. The adoption of a SOA must not be considered a technological necessity but rather an organized one: it must respond to the needs of the organization.

- Managing the process. The services and applications that correspond to the desired information processes and outputs through the diverse functional areas of the organization.

For an implementation of a SOA it is necessary consider four phases, realizing various interactions in each one of them [Figure 4.3] [1].

4.6 SOA applications in the business world

The following are business applications where SOA applications have been implemented successfully:
4.6.1 Sending automation of client’s requests.

Northern Electronics is a manufacturer of electronic components with its headquarters in Everett, Washington. Such company suffered from different competitive pressures and looked for a way to increase its advantage through a more efficient management, by integrating and automating the processes to eliminate the inefficiencies within the process of sending of clients’ requests, where they opt for a SOA who could contribute by creating within Northern Electronics an infrastructure that is flexible, technologic, and well-interconnected.

An layer oriented to flexible and reusable services who exposed a business logic of the actual applications (and that of their partners) was implemented, with the purpose of reducing the detected inefficiencies in the manual request processes and state notification. This layer of services was implemented by using their own developments based on .NET Framework and Visual Studio as well as external integration adaptors for their applications of business line and mainframe systems.

The first implementation project was limited in an explicit way and its return of investments were obtained within ninety days. Even after the SOA infrastructure finished, Northern Electronics has continued to capitalize these benefits. Without having to wait for a great implementation project of infrastructures that last various years, they can now, almost instantly, manage their value chain in a more efficient way. Furthermore, they are capable of identifying and taking advantage of new opportunities to consolidate and to automate internal activities within their value chain[17].
4.6.2 Process automation in securing activities.

Mapfre is an independent Spanish business group that offers transportation insurance in forty countries. Mapfre acquired the software application Crossvision SOA/BPM so that, from their actual systems, each Mapfre international branch can automate the critical processes that are considered more relevant for the local market (hiring, management of a disaster, management of a commercial web, etc.) guaranteeing a significant improvement in the efficiency of the processes, greater service quality, and greater client satisfaction.

All in all, the Crossvision SOA/BPM will give support to more than 4,000 users in the international insurance branch directly from Mapfre located in Latin America, who will benefit from this application for the modeling, automating, and optimizing of their critical processes [37].

4.6.3 Grid Computing - Condor

The grid community has been migrating towards service-oriented architectures as a way of exposing and interacting with computational resources across organizational boundaries. The SOA architecture was adopted by the Condor Project in order to increase the level of manageability, extensibility and interoperability between loosely coupled services that are crucial to the development of a grid infrastructure spanning multiple organizations and incorporating a wide range of different services. Multiple support was created and adapted into the Condor System, including Web services to ensure open interoperability with future mainstream grid developments. By interpreting the Condor resources as services, they can be easily adapted into this emerging service oriented environment: allowing third parties to incorporate Condor’s capabilities into their own applications, and considerably improving upon Condor’s ability to operate across organizational boundaries[13].

4.7 Summary

The chapter has exposed SOA as well as technologic implementations of it. It also talked about the benefits of using SOA if implemented correctly. The theoretical basis that SOA proposes helps understand what Grid Computing must accomplish in order to better integrate the Business Process of the VO with the proper service applications. The following chapters will describe Grid computing and the implementation used to approach the ICT-I functionality found in the creation and operation of VOs.
Chapter 5

Grid Computing

The applications’ services of each enterprise need to be integrated to the corresponding business process in an organized manner. Grid Computing is being suggested to serve as that integrator technology. This dissertation uses Grid Computing as the ICT infrastructure that supports the creation and execution of the VO processes. This chapter will describe Grid Computing and its role in the research.

5.1 What is Grid Computing?

In 1998, Ian Foster and Carl Kesselman defined a computational grid as [26]: “a hardware and software infrastructure that provides dependable, consistent, pervasive and inexpensive access to high-end computational capabilities.” Such definition does not apply today because it has become insufficient due to the evolution of the concept. Today, Grid Computing is used much more widely and its scope has widen as well, thus, it has created ambiguity and raises questions about Grid Computing that could be easily answered if only there were some sort of protocol that everyone could use when using Grid Computing.

The need for having a Grid Computing protocol is necessary to be able to set up a method where the different individuals can come to an understanding about how, what, and to whom is it that they want to share and what work they want to undertake. So, to know what Grid Computing is, one needs to understand what Grid Computing does [25]:

- It offers an Infrastructure to coordinate resources that are not subject to centralized control (but is not limited to).
- It reassures that the resources can be depended on.
- It offers consistency on services by working according to standards.
- It offers pervasive access, making sure that the services will be available at all times in various environments.
Grid computing provides a large amount of computational capabilities from a wide collection of connected systems sharing various combinations of resources. It serves as the manager of all the processes involved within a particular task, by collecting the available small, medium and huge computer resources (including mainframes and their respective software), and makes use of them to get the job done in lesser time and with less money[26]. The same task, done without a Grid Architecture, would require a great investment in hardware and software or would take normal computers ages to finish. It is when a task is so large that Grid Computing can achieve its full potential: by dividing the task and distributing the pieces to the various resource entities, thus, overcoming it by working “hand in hand”. [62].

5.2 Underutilized Computing Resources

The simplest way in which grid computing can be used is to run an existing program on a different machine. The idea is to take advantage of the idle period of machine resources within the grid, by remotely executing the application required and without undue overhead. For this it is important that the remote machine meets any special hardware, software, or resource requirements imposed by the application.

Tasks, which require a significant amount of time to process an input data to produce an output set, are perhaps the most ideal and efficient use for a grid [22]. There are large quantities of wasted computing resources in most organizations. According to various studies, academic and commercial entities report utilisations of around 30%, and in many cases computers and servers are busy less than 5 percent of the time[26]. Grid computing provides a framework to take advantage of them and maximize the efficiency of resource usage.

Another issue in which grid computing can be taken advantage of is unused memory storage. A grid can be used to convert this dispersed unused storage into a much larger virtual memory source, and at the same time, increase performance and reliability over that of any single machine.

Not only can Grid Computing rescue unused resources, but it may also help to balance resource utilization (see figure 5.1). If a computer receives an unexpected amount of activity, some grid implementations can migrate partially executed jobs to other parts of the grid where an underutilized machine can continue the task. Such balance can be focused on CPU use, storage, and other kinds of resources that may be on a grid. The use of Grid Computing eliminates the need to view the usage patterns in the larger organization, as well as to allow better planning when upgrading systems, increase memory capacity, retire computing resources, etc.

Other possible underutilized resources could be software, licenses, bandwidth, different types of printers, DVD writers, special technical equipment, robots and the list goes on and on. The additional resources can be provided in additional quantities, speeds and capacities. The resources provided by such grid would be far greater than what a conventional desktop computer could provide.

This last issue of “resource balancing” is very similar to the issues found in CNO
Figure 5.1: Jobs are migrated to less busy parts of the grid to balance loads [22]

supporting functionality “Dynamic reconfiguration of the VOs” where members of the VO (representing resources) can be effectively interchanged to reduce workload and stand-by time.

5.3 The Components of a Grid Computing

To create a proper Grid Computing environment there need to be several capabilities (which are subcomponents) that Grid Computing must be able to accomplish, and there is a complex array of capabilities [7]:

- Remote storage and/or replication of data sets.
- Publication of datasets using global logical name and attributes in the catalog.
- Security access authorization and uniform authentication.
- Uniform access to remote resources (data and computational resources.)
- Publication of services and access cost.
• Composition of distributed applications using diverse software components including legacy programs.

• Discovery of suitable datasets by their global logical names or attributes.

• Discovery of suitable computational resources.

• Mapping and Scheduling of jobs (Aggregation of distributed services.)

• Submission, monitoring, steering of jobs execution.

• Movement of code/data between the user desktop machines and distributed resources.

• Enforcement of quality of service requirements.

• Metering and Accounting of resource usage.

The Grid components that provide all of those capabilities must be arranged into four essential layers. The layers are organized in levels from basic services/tasks to more complex ones. Those four main components are (beginning from the most basic) [7]:

1. **Grid Fabric**: layer consists of distributed resources such as computers, networks, storage devices and scientific instruments. The computational resources represent multiple architectures such as clusters, supercomputers, servers and ordinary PCs which run a variety of operating systems (such as UNIX variants or Windows). Scientific instruments such as telescope and sensor networks provide real-time data that can be transmitted directly to computational sites or are stored in a database.

2. **Core Grid**: middleware offers services such as remote process management, co-allocation of resources, storage access, information registration and discovery, security and aspects of Quality of Service (QoS) such as resource reservation and trading. These services abstract the complexity and heterogeneity of the fabric level by providing a consistent method for accessing distributed resources.

3. **User-level Grid**: middleware utilizes the interfaces provided by the low-level middleware to provide higher level abstractions and services. These include application development environments, programming tools and resource brokers for managing resources and scheduling application tasks for execution on global resources.

4. **Grid applications and portals**: are typically developed using Grid-enabled programming environments and interfaces and brokering and scheduling services provided by user-level middleware. An example application, such as parameter simulation or a complex math computation, would require computational power,
access to remote datasets, and may need to interact with scientific instruments. Grid portals offer Web-enabled application services, where users can submit and collect results for their jobs on remote resources through the Web.

Figure 5.2 illustrates the four main components [7].

![Layered Grid Architecture and its components](image)

**Figure 5.2: A layered Grid Architecture and its components**

### 5.4 Grid Components and the ICT Infrastructure used in by VOs

The interoperability issues needed by the ICT Infrastructure to be useful in the Operation and Execution of a Virtual Organization are listed on table 5.1 with the corre-
sponding Grid subcomponents that support it.

<table>
<thead>
<tr>
<th>ICT Interoperability Issue</th>
<th>Grid Subcomponents that Support it</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Access from multiple Devices</td>
<td>Since the grid has both a Middleware and an Applications Portal layer, it is possible to create access to the resources through the internet across multiple machines and multiple operating systems.</td>
</tr>
<tr>
<td>2. Support for Business Collaboration</td>
<td>The enterprise services (resources to the grid) need to be well documented to be configured into the Grid system, once integrated there can be communication between broker and each of the enterprises to a certain degree.</td>
</tr>
<tr>
<td>3. Seamless Integration of Services and Applications</td>
<td>Once integrated to the grid system, the broker can manage which jobs will go to the most appropriate enterprises and if a certain enterprise is unavailable, the grid can look for another that matches the same capabilities for the job.</td>
</tr>
<tr>
<td>4. Heterogeneous Applications Integration</td>
<td>Since the grid can work over the internet, it can work across the world and across multiple operating systems that support the grid, taking care of the distance and language barriers.</td>
</tr>
</tbody>
</table>

Table 5.1: Grid ICT Functions and the Grid Subcomponents that support it (adapted from [11],[7])

### 5.5 Comparison with Cloud Computing

Cloud Computing is “the sharing of computer resources and services over the internet”. This can reduce enterprise expenses as well as reducing installation and implementation time. Most Cloud Computing services use Grid computing for resource management, so a “Cloud can be a Grid” but not the other way around[18]. The Cloud Computing buzz word has become popular recently and some grid computing vendors changed their name to cloud computing to get more business. The concept is relatively new and after going through many name changes (including the name “Grid”), it finally
established itself [52]. This experiment does not share computer resources, but the enterprise resources which are published and managed through the Grid.

5.6 Applications

There are many factors to consider in “grid-enabling” an application. One must understand that not all applications can be adapted to run in parallel on a network to achieve scalability. Furthermore, there are no practical tools for transforming arbitrary applications to exploit the parallel capabilities of a grid. There are some practical tools that skilled application designers can use to write a parallel grid application. However, automatic transformation of applications is a science in its infancy. This can be a difficult job and often requires top mathematics and programming talents, if it is even possible in a given situation. New computation intensive applications written today are being designed for parallel execution and these will be easily grid-enabled, if they do not already follow emerging grid protocols and standards. The following are some of Grid Computing’s successful applications.

5.6.1 Large Hadron Collider

Grid is very useful for the exploitation of large sets of data which are currently not enough exploited due to a lack of tools, for running simulations and for statistical approach used for event prediction. Grid emphasizes the collaboration among various scientific and technical domains, a very fruitful point [19]. One application of grid is found in the large community of Large Hadron Collider (LHC). LHC is (as its name implies) a large atomic particle accelerator built as a collaboration among more than 50 countries [19]. Thousands of scientists and institutions have invested a large amount of time and money in this accelerator because it would provide the high-energy physics community crucial information from questions about fundamental particles to dark matter and dark energy (the remaining portion of the world that is not made up of matter) and even the Big Bang theory [23].

Many sub-experiments are at work using the LHC for the data that is generated by it: the ALICE, the ATLAS, the LHCb, to name a few. All of the projects are working simultaneously (and, of course, are globally dispersed) therefore, the role Grid Computing plays in LHC’s projects is that of a computational infrastructure that couples wide-area distributed resources like databases, storage servers, supercomputers, etc. to solve massive problems [7]. Although there might have been other infrastructures that could have been used, Grid Computing was chosen primarily because it was the most inexpensive. From the beginning of the LHC managing project it was obvious that a free or cheap way of obtaining huge computer capacity was needed. So the collaborators of the LHC integrated their own computing facilities to have one huge computing service; the LHC Computing Grid [24].
5.6.2 Physics and Earth Science

Grid Computing has been also applied for research in Gravitational Physics. Grid Computing was used to simulate the gravitational effects of black hole collisions [4]. Yet another project was the Monitoring System for the Earth System Grid which was used by scientists who study the Earth’s climate. Grid Computing was used because the project needed large datasets that would only be able to be generated by massive computational power [60].

5.7 Implementations

In order to make Grid Computing work, it needs support of software. Globus[4], Condor[58], Sun Grid[53] and GridGain[55] are just some of the many Grid Computing enabling software that have been developed. The following list briefly describes the mentioned frameworks:

- **Globus**: a project that was developed by various universities, enterprises and public administrations of the U.S. It consists of a series of components that proportionate basic services, such as security, information and communications [2].

- **Condor**: (originally a system designed to utilize CPU cycles of tedious work stations) is a workflow editor and scheduler made from the “Condor Research Project” of the University of Wisconsin-Madison [58].

- **SunGrid**: like Condor, is a commercial licence workflow editor and scheduler.

- **GridGain**: is a Java based API for Grid Computing. It is an open source product that appeared in July of 2007. The key functions are Map/Reduce and Service Provide Interface (SPI) [2].

Table 5.2 shows the advantages and disadvantages of each Grid Implementation, as well as the list of the ICT Interoperability issues it can support for the VO Process (the actual support magnitude would require an in-depth analysis) [42].

Of all the existing Grid Computing implementations Condor was chosen due to being open-source and its independence from a shared file system across machines. It was also chosen for its ease of use and portability across various platforms and configurable process broker (task manager). The configurable process broker is essential for the functionality required during the orchestration and execution of the processes during the operation of a VO. The next chapter will discuss the Condor Grid System in detail.
<table>
<thead>
<tr>
<th>Implementation</th>
<th>ICT Interoperability Support</th>
<th>Benefits</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| Globus Toolkit | - Heterogeneous Applications Integration  
                 - Seamless Integration of Services  
                 - Applications and Security Integration | - No licence costs (open source)  
                 - Best Security Infrastructure  
                 - Support from big Companies due to its maturity | - Less support on Windows  
                 - Requires a resource dedicated to Task Management  
                 - Is not a “turnkey” solution, it is a toolkit that requires exhaustive setup |
| SunGrid       | - Business Collaboration Support  
                 - Heterogeneous Applications Integration  
                 - Seamless Integration of Services and Applications  
                 - Security Integration | - Specialist technical support  
                 - Complete solution out of the box  
                 - Interoperability Across many platforms | - Licencing cost  
                 - Can be tied into single vendor |
| Condor High Throughput computing Scientific Research | - Business Collaboration Support  
                 - Heterogeneous Applications Integration  
                 - Seamless Integration of Services and Applications  
                 - Security Integration | - Easy installation across multiple platforms  
                 - Open Source  
                 - Robust functionality  
                 - Strong and Flexible matchmaking capabilities | - Requires coordination between sites  
                 - Security is less mature in comparison with other vendors |
| GridGain      | - Business Collaboration Support  
                 - Heterogeneous Applications Integration  
                 - Security Integration | - Open Source  
                 - Development in Java language supports interoperability across many operation | - Task managing capabilities are very limited and not configurable  
                 - Requires a lot of setup to configure scenario |

Table 5.2: Grid Software: advantages and disadvantages (adapted from [42], [58], [2])

### 5.8 Summary

The chapter has described what Grid Computing as well as its capabilities. Some of the Grid Computing capabilities were explained to be able to handle the Interoperability Issues needed for the ICT Infrastructure. These issues are similar to the issues mentioned in the BPM Chapter (System Interoperability and Dynamic Data Exchange). Grid Computing can handle most of the issues to a basic level. Summarizing, the VO broker can use grid computing to interpret each business as a “resource” and integrate each application service through it, this way, it can manage the Collaboration Process as a Grid Computing job and find the proper “resources” to resolve them.

The grid computing system chosen was Condor due to its portability and ease of use as well as its functionalities being closely related to the functionality required for
the creation and operation of Virtual Organizations. The next chapter will describe the Condor Grid System and the components used for this dissertation.
Chapter 6

Condor

Now that Grid Computing has been defined, the Condor High Throughput Computing Project was chosen to create a prototype of a Collaborative Process scenario in the VO creation and execution steps. As mentioned before, the implementation requires to solve some specific issues:

- Interoperability between different platforms and across the Internet.
- Ability to define resources and manage them dynamically
- Ability to define a process
- Ability to manage and document a process and the resources involved.

This chapter attempts to provide an introduction to this freely available and robust tool in order to clarify why it was selected for implementing the Collaborative Process in the VO.

6.1 What is Condor?

The Condor development and research team describes it as [44] “A reliable, scalable, and proven distributed system, with many years of operating experience. It manages a computing system with multiple owners, multiple users, and no centralized administrative structure. It deals gracefully with failures and delivers a large number of computing cycles on the time scale of months to years.”

Condor is specialized for workload management and coarse-grained distributed parallelization of computationally intensive jobs. It can be used to manage workload on a dedicated cluster of computers, and/or to farm out work to idle desktop computers(cycle scavenging). Condor runs on Linux, Unix, Mac OS X, FreeBSD, and contemporary Windows operating systems(it also has legacy application adaptors). Condor can seamlessly integrate both dedicated resources (rack-mounted clusters) and non-dedicated desktop machines (cycle scavenging) into one computing environment.
6.2 How does Condor Work?

Condor provides a task management mechanism, scheduling policy, priority scheme, resource monitoring, and resource management. Users submit their jobs and the system chooses when and where to run them based upon a policy, monitors their progress, and informs the users upon completion. The jobs submitted are described in a text file with the extension “.submit”. [59].

6.2.1 Condor Components

Figure 6.1 displays the interaction between the basic components that make up Condor.

- The **Schedd** is client of Grid computing services and is the entry point for end users. It provides a transaction interface for submitting, querying, and removing tasks, and is responsible for storing the tasks while it finds where it can match the requirements of execution and to execute them. The Schedd is responsible for enforcing user requirements on job execution. For example, the user may require that jobs only run on machines that have sufficient memory and that are owned by a trusted user.

- The **Startd** is a Grid computing service which manages a particular execution machine, and is responsible for finding work to be done that fits the constraints.
defined by the machine’s owner and establish the priority of tasks which meets such restriction.

- The **Matchmaker** has to identify compatible consumers (Schedds) with producers (Startds). It accepts advertisements from all parties, written in ClassAd description language. Once a potential match is detected, both resources are notified. Among other responsibilities, the Matchmaker controls admission to the pool, the number of machines allocable to any user, and monitors pool-wide resource usage.

- The **Starter** is in charge of managing the execution environment at the remote site. It converts a job description into the operating system calls that actually execute and monitors the job for its exit status. The Starter informs the other Condor components whether or not the job was able to execute.

- The **Shadow** is the component responsible for making all policy decisions needed by a job. The Shadow prepares the executable, arguments, environment, standard I/O streams, and everything necessary for a complete job specification. When the job finishes, the Shadow is responsible for determining if the job was successfully run by analyzing the exit code of the job, the output data, the execution machine, and any other relevant information. It is also responsible for establishing the I/O channel that may be used to fetch the executable from the home site and to perform online input and output.

### 6.2.2 The Matchmaking Framework

Condor implements matchmaking by allowing entities which provide or require a service to advertise their characteristics and requirements in classified advertisements (Classads). A Matchmaking service then matches classads in a manner that satisfies the constraints from the respective advertisements and informs the interested entities of the match. The matched entities communicate and cooperate to perform the desired service. The Matchmaking framework [51] may be decomposed into five components (see figure 6.2):

- The **Classad specification**, which defines a “language used by Condor to advertise resources and requests for those resources in a distributed environment” [14]. An advertisement, called a **ClassAd**, represents an entity advertising itself or requesting for a resource and consists of named descriptive attributes, constraints, preferences and semantics of evaluating these attributes.

- The **Advertising Protocol** defines basic conventions in terms of what a matchmaker expects to find in a classad and how the matchmaker expects to receive the Ad from the advertiser.

- The **Matchmaking Algorithm** establishes how the properties of Ads and the state of system relate to the outcome of the matchmaking process.
6.3 Classified Advertisements (ClassAds)

In [54], the authors compare ClassAds to the classified advertisement section of a newspaper, where sellers advertise specifics about what they have to sell, and buyers may publish specifics of what they wish to obtain. In both cases certain characteristics or constraints must be satisfied, like price limits. In a similar way, Condor users submitting jobs that can resemble buyers of computing resources and machine owners sellers. A Classad is a flexible and extensible data model that can be used to represent arbitrary services and constraints on their allocation. Classads use a semi-structured data model, so no specific schema is required by the matchmaker, allowing the matchmaker to work naturally in a heterogeneous environment.

Constraints (i.e., queries) may be expressed as attributes of the classad. Classads are first-class objects in the model. They can be arbitrarily nested, leading to a natural language for expressing resource aggregates or co-allocation requests. A classad is a mapping from attribute names to expressions. All computers in a Condor pool use classads to describe themselves in terms of their attributes, such as RAM, CPU speed, disk space, Operating system, software installed and any other resource they might
have access to.

A ClassAd is an aggregate of the following information[36]:

1. Type information. This is a tuple of 2 types, self type and match type, which characterizes the kinds of entities involved in the match.

2. Attribute List. A list of arbitrary attributes which describe the properties of the advertising entity.

3. Requirement. An expression which represents the constraints that the advertising entity places on the match Ad.

4. Rank. An expression representing the advertising entity’s assesment of the quality of a match.

5. Advertisement ID. If an entity publishes several Ads and one match is found, then the match can be qualified with an advertisement ID of the matched Ad so that the advertising entity can identify which Ad was matched.

Figure 6.3 shows a classad that describes a workstation in a Condor pool at the University of Wisconsin, where any of the attributes in a machine Ad can be utilized at job submission time as part of a request or preference on what machine to use. Additional attributes can be easily added and attributes may be simple integer, real, or string constants, or they may be more complex expressions constructed with arithmetic and logical operators and record and list constructors. This last issue is important for the definition of enterprises since their non-scientific capabilities can be defined as resources.

6.4 Matching

Entities and resources advertise themselves to a matchmaker, which is responsible for introducing compatible agents and resources. The matchmaker is also responsible for enforcing community policies such as admission control. The matching process basically consists of the four steps shown in figure 6.4.

Classads are created describing the machines in the pool and submits them to the Matchmaker (Step 1). These classads are constructed to conform to the advertising protocol specified by the matchmaker, which attaches a meaning to some attributes. The advertising protocol also specifies how the entities send the classads to the matchmaker. The matchmaker then invokes a matchmaking algorithm by which matches are identified (Step 2). After the matching phase, the matchmaker invokes a matchmaking protocol to notify the two parties that were matched (Step 3) and sends them the matching Ads. The matchmaking protocol could also include the generation and hand-off of a session key for authentication and security purposes. The customer contacts the server directly, using a claiming protocol to establish a working relationship with the provider (Step 4). Identifying a match and invoking the matchmaking protocol
does not immediately grant service to a customer but it just introduces the advertising entities. The state of service providers and requesters may be continuously changing, thus allowing a possibility that the matchmaker made a match with an obsolete ad-
advertisement. The provider and customer verify their constraints with respect to their current state.

6.5 DAGman

The Directed Acyclic Graph Manager (DAGMan) is a service for meta-scheduling multiple jobs with dependencies in a declarative form\[45\]. It accepts a declaration that lists the jobs to be executed with constraints on the order and dependency of output of other jobs. DAGMan keeps private logs to detect completion of jobs, thus, allowing it to resume a DAG where it left off, even in the face of crashes and other failures \[57\]. Workflows are represented as directed graphs where each node denotes an atomic task and the directed edge indicates a dependency relationship between two adjacent nodes\[48\]. Figure 6.5 shows an acyclic graph and the language used by DAGMan to model it.

![Figure 6.5: A Directed Acyclic Graph\[59\]](image)

In the figure 6.5, the JOB A statement associates an abstract name (A) with a file (A.submit) describing a complete job. A PARENT-CHILD refers to the relationship between two or more jobs. In this example, jobs B and C may not start executing until A has completed, and jobs D and E wait for job C’s completion. When jobs do not need results of each other they may run in any order and even simultaneously. The PRE and POST commands refer to programs which need to be run before and after a job executes and such programs are run by DAGMan on the submitting machine. PRE jobs are usually employed to prepare the execution environment, while POST programs are generally used to evaluate the output of the job \[59\].

There could be many reasons to cause a job to fail. Network problems or resources needed by a higher priority process could cause Condor to lose contact with a running job. These, however, are not indications that the job itself failed, but that the system has failed and DAGMan is never aware of this. But if a job produces an ordinary error result, such as a bad input or output DAGMan is aware of such error. A rescue DAG
is a new DAG containing the elements of the original DAG that were not executed and may be re-submitted to continue the process. The POST could help analyzing script failures, triggering a RETRY or the production of a rescue DAG.

6.6 Real World Applications of Condor

Condor is being currently used by several organizations around the world to solve many different types of problems. Below are described a few examples of the way Condor is being employed.

6.6.1 Planets Prefer Wacky Orbits

A project by Kenneth Chang[12], at the National Center for Supercomputing Applications. Condor was used for simulations of planetary orbits. “On a single workstation, the calculations would take forever. Here we can do thousands of integrations on the order of weeks. It’s a very nice system.” In this project, thousands of simulations need to be run with different initial conditions and then examine the statistical probabilities of each outcome. All those simulations consume a lot of computer time, so they take advantage of the Condor computing system to distribute simulations across an existing network of university workstations, taking advantage of otherwise idle time on the computers.

6.6.2 C.O.R.E. Digital Pictures

C.O.R.E. Digital Pictures is a Toronto-based computer animation studio. Photorealistic animation is a computing intensive process, where each frame can take up to an hour, and one second of animation can require 30 or more frames. C.O.R.E employs a Condor pool of hundreds of Linux and Silicon Graphics machines. The Linux machines are all dual-CPU and mostly reside on the desktops of the animators. By taking advantage of Condor ClassAds and native support for multiprocessor machines, one CPU is dedicated to running Condor jobs while the second CPU only runs jobs when the machine is not being used interactively by its owner. C.O.R.E. animators submit over 15,000 jobs to Condor.

C.O.R.E. makes considerable use of the schema-free properties of ClassAds by inserting custom attributes into the job ClassAd. These attributes allow Condor to make planning decisions based upon real-time input from production managers, who can tag a project, or a shot, or individual animator with a priority. Condor has been used by C.O.R.E. for many major productions such as X-Men, Blade II, Nutty Professor II, and The Time Machine.

6.6.3 NETCARE

NETCARE [59] uses Condor to enable electronic design automation. Its foundation is a partnership between two well-established systems - PUNCH from Purdue University
and Condor. NETCARE-managed resources will include dedicated machines as well as a large number of opportunistic resources (i.e., Condor-managed desktop workstations and servers whose idle cycles are used for NETCARE jobs). PUNCH provides the user interface, manage the repository of software tools and performs high-level scheduling and dedicated resource access-control. NETCARE implements low-level scheduling and management of resources through Condor.

6.6.4 Condor high-throughput DVD transcoding system for Linux.

Because much multimedia content is available as DVD MPEG-2 files, it is necessary to transcode them to obtain the MPEG-4 equivalents. In [28], the authors propose a Linux framework based on the Condor platform to achieve high-throughput DVD transcoding. Although some LAN parallel transcoding tools for fixed sets of machines exist, there did not exist one for parallel transcoding. Metacomputing refers to architectures that hide physical resources and instead offer a simplified virtual machine view. For example, the Condor tool “steals” cycles of available machines when neither users nor high-priority processes are using them.

6.7 Summary

This chapter has described the Condor Architecture and described the various components it contains to be able to approach the functions needed for the creation and operation of Virtual Organizations. To recap the components and how they will be used we have:

- Condor Components: These components will be used to define each section of the business process, the requirements needed for its execution and the tools to send it to its proper executor. It also documents the whole job.

- Matchmaking: The tool used to choose the proper enterprise for the process at hand.

- ClassAd: The way each enterprise can be defined as a resource, facilitating the Virtual Broker decision for the best entity for a Job.

- Dagman: Each section of the business process can be unified through a Directed Acyclic Graph thus creating the model of a whole business process in which several processes can be executed in parallel.

The next chapter will describe the case study to be implemented in Condor to demonstrate the potential utilization of Grid Computing in general and the Condor framework in particular for the functionalities required to manage the VO Collaboration Processes.
Chapter 7

CASE STUDY

This Chapter describes the case study scenario that will be used within the Condor System, this particular case scenario is a fictitious Collaborative Software Development Project (to which we will refer to as CSDP) based on actual software development processes used by Open Source Companies like FreeBSD, OpenOffice.org, and Java as well as commercial companies like IBM and Microsoft. After describing the scenario, the implementation of the grid framework will be shown to demonstrate the contribution to this research project[21][30][49][46].

7.1 The Collaborative Software Development Project - CSDP

This case study consists of a Collaborative Software Development Project in which various software developing companies around the world contribute to the same goal. We are going to refer to this group of companies as the “Pool of Contributors.” All companies within this Pool have different capabilities and levels of expertise (figure 7.1).

The Project Broker is to distribute the developing and testing processes among the Pool of Contributors. The companies are called by the Broker when their area of expertise is needed for a particular process.

The project has already begun and a Virtual Organization composed of many contributors from around the world has already been assembled for this particular project.

7.2 Project “Free Office Suite”

The project “Free Office Suite” is the current software being developed by the CSDP. This is a robust software composed of various modules that work in unison to provide different office related services to a user. Once finished, the modules are compiled and tested together for the final build.
Figure 7.1: Collaborative Software Development Project Creation

The Free Office suit is composed of the following modules:

1. Writer: A word processing software.
2. Impress: Presentation creation software.

These five separate modules can be built and tested independently from one another, in other words, the modules can be worked on in parallel to one another. A model of the building process is shown in the figure 7.2:

The modules have already been developed by this time and the jobs that are left are Debugging and Testing before the final build. The following are descriptions of the processes left to work on:

1. Build Review: The “Free Office Suit” is checked as a whole for updates or errors in the current modules and a specification document of these changes is created.

2. Module Debugging: The list of changes is received by each of the modules and the debugging takes place. Since all modules are independent from one another, this process can be done in parallel with the rest.

3. Module Testing: After the debugging, the changes need to be tested. This process can also be done in parallel and it doesn’t have to be done by the same entity that did the debugging.
4. Build Approval: Once all of the modules have been tested the final build can be assembled for a final check.

All of the described processes have to be completed by different companies that belong to the contributor’s pool. Most of the companies that perform debugging and testing are located in places distant from one another, thus organizing many companies to work and communicate together can be quite a challenge. Table 7.1 contains the company names and location.

Each of the companies have different capabilities and areas of expertise and are essential to the project, and maximum coordination between these companies needs to take place in order to work efficiently. Although all of the companies belong to the contributor pool, they all don’t need to be working on the same project at a time. There can be cases when a company in the pool is not available to participate. There are also cases when new companies should form part of the pool because of a certain expertise. We can conclude that the contributor pool is subject to dynamic changes.

Another obstacle in this scenario is that the current BPM software does not support collaboration efforts among companies, in other words, the current technology is focused on orchestrating processes of only one company. A tool with process orchestrating capabilities over a large network between interchangeable entities is needed. This is why a Grid Computing System is proposed.
Table 7.1: Enterprises and Their Locations

<table>
<thead>
<tr>
<th>Name of Contributor</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company Zero</td>
<td>Houston, Texas, USA</td>
</tr>
<tr>
<td>Company One</td>
<td>Mumbai, India</td>
</tr>
<tr>
<td>Company Two</td>
<td>Hong Kong, China</td>
</tr>
<tr>
<td>Company Three</td>
<td>Monterrey, Mexico</td>
</tr>
<tr>
<td>Company Four</td>
<td>Guadalajara, Mexico</td>
</tr>
<tr>
<td>Company Five</td>
<td>Houston, Texas, USA</td>
</tr>
<tr>
<td>Company Six</td>
<td>D.F., Mexico</td>
</tr>
<tr>
<td>Company Seven</td>
<td>Dallas, Texas</td>
</tr>
</tbody>
</table>

7.3 Setting up the Grid Network

To make the case study function as intended, the Condor Grid Software had to be configured to fit with the nature of the problem. The Condor Grid tool was installed on 3 Virtual Machines via VMware Workstation Software. The Virtual Machines will simulate to be the VO members as well as the VO Broker. This test was limited to 3 Virtual Machines, any more would have caused a great workload on the computer it was implemented on. The machines will prove enough to demonstrate the capabilities of the grid.

The Virtual Machines were running Windows XP, with Java Virtual Machine 1.6 and The Condor Grid Engine installed. All of the machines belonged to the same Grid Pool, enabling them to communicate with one another. Each of the machines had job-running capabilities but only the broker machine had job-submitting capabilities.

The Virtual Computer Setup is like this:

<table>
<thead>
<tr>
<th>Computer Name</th>
<th>Company</th>
<th>Permission</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broker</td>
<td>Company Zero</td>
<td>Job Submitting and Running</td>
</tr>
<tr>
<td>Client1</td>
<td>Company 1</td>
<td>Job Running Only</td>
</tr>
<tr>
<td>Tester</td>
<td>Company 2</td>
<td>Job Running Only</td>
</tr>
</tbody>
</table>

Table 7.2: List of Entities in the grid setup

7.4 DagMan Definition

The virtual computers will take the roles of the “contributor enterprises.” The actual collaborator process model will be interpreted as a Directed Acyclic Graph (DAG) defined in the Condor Grid. The process model can be represented as a series of
computations in which their input depends on the output of other computations (see figure 7.3).

Figure 7.3: DAG model

Condor’s DAG is defined in the following text:
Software VOs

JOB BuildReview BuildReview.Review.submit
JOB WriterMod Writer.Debugging.submit
JOB CalcMod Calc.Debugging.submit
JOB ImpressMod Impress.Debugging.submit
JOB DrawMod Draw.Debugging.submit
JOB WriterTest Writer.Testing.submit
JOB CalcTest Calc.Testing.submit
JOB ImpressTest Impress.Testing.submit
JOB DrawTest Draw.Testing.submit
JOB BuildApproval BuildApproval.Testing.submit

PARENT BuildReview CHILD WriterMod CalcMod ImpressMod DrawMod
PARENT WriterMod CHILD WriterTest
PARENT CalcMod CHILD CalcTest
PARENT ImpressMod CHILD ImpressTest
PARENT DrawMod CHILD DrawTest
The SoftwareVOs.dag file is the DAG definition for the Condor Grid to execute and manage between the appropriate computer/contributors. This file lists the tasks to be done and the appropriate order in which they are to be executed. Some tasks can be executed in parallel as long as there is an appropriate collaborator available to take responsibility of the task. Notice how there is just the listing of the jobs and the order in the DAG file, the definition of the jobs is defined separately.

### 7.5 Job Definition

Each one of the tasks listed in the DAG file have to be defined individually in “job submit” files. As mentioned previously, a submit file contains all the information necessary for a task to be submitted and executed in the Condor Grid. In addition to the description of the task, we have also added fictitious requirements to make the Condor Broker search for the proper computer/contributor that can meet those requirements. The following shows the submit file for the Base Testing job:

Base.Testing.submit

```
Universe = vanilla
Executable = c:\condor\bin\prueba\Executer.bat
Arguments = BaseTesting.txt $(Cluster).txt
Requirements = Name == "Client1"
Output = c:\condor\bin\prueba\LOGS\BaseTesting.out
Error = c:\condor\bin\prueba\LOGS\BaseTesting.err
Log = c:\condor\bin\prueba\LOGS\BaseTesting.log
when_to_transfer_output = ON_EXIT
getenv = true
Queue
```

In this particular case, the job requires for a Computer/Contributor to have a Windows operating system and to be named “Client1.” Many different types of requirements can be added and are not limited to computer hardware, like the original Condor intentions. In order to use personalized non-computer requirements like “Man-power ≥ 200” or “JavaProgramming = 1,” they must be defined in the configuration files of every computer within the pool. The arguments used are described in two text files with the job details in them. Two arguments are used in order to document which entity has control of the job during execution. The full list of submit files are presented in Appendix B.

The requirements for each submit file are varied in order for the Condor broker to demonstrate its capabilities and choose the proper person for each job. All of the submit files log their execution history individually. The common text that all submit files share is the “Executable.” The file executed by each process is a console batch file.
This batch file is used to manage the invocation of a C++ executable program located on each target computer. The batch file can receive multiple parameters and run the executable for each one received or return an error if no parameter was received. A description of the executable is discussed in the next section.

7.6 The Executable

The Condor Grid was designed to run computer applications that usually involve some sort of scientific calculation. In this particular scenario, no calculations need to take place within the computer, but from the contributor itself. The computer receiving the job will act as a medium to communicate with the actual Contributor Enterprise. In order to simulate the communication with the broker and each of the enterprises, a simple C++ program was created to act as the scaffold of an enterprise application.

The program takes the parameters from the submit file (in this case, the actual specifications of the task) and stores them within a text file which is created in the computer owned by the Contributor Enterprise. Once the text file with the task information is created, another C++ program running within the Contributor computer detects the program and acknowledges the task as it is being received. The first program waits for the information to be received before ending. This is done to simulate an actual enterprise receiving the specifics of a certain job and having the choice of rejecting or accepting the responsibility to get it done. In order to make the test run smoothly, the executable is programmed to always accept the jobs. It is important to understand that even though the communication done in this simulation is instant, the real life process does not work as quickly as it is shown. In real life, each enterprise would take more time responding to the request done by the broker, as well as take more time doing the actual job.

The source code of the two C++ programs is located in Appendix C. Now that the individual parts of the execution have been described the actual test run is described in the following section.

7.7 Experiment Run

The details of the test run are described in this section. The details of all the run are logged by Condor, and full detail of the output is located in Appendix D. Both the Condor job submission and monitoring of the task is done via console. The first step for the test run is to make sure that all computers are ready and are part of the same pool. This is done with the CONDOR_STATUS command, see Figure 7.4:

The CONDOR_STATUS command shows all of the computers that are part of the pool with the exception of the Condor Central Manager, in this case the computer “Broker” is the Condor Central Manager. The command lists the technical details of the computers in the pool, and if they are currently working on any jobs. The relevant attributes described are:
1. Name: The name of the computer.

2. OpSys: Operating System found in the computer.


4. State: Denotes if the computer has been asked to do a job.

5. Activity: Denotes if the computer is currently working on a process or available.

The current Activity attribute shown is “idle” which means the computers are not currently working on any jobs and are available to receive jobs.

The next step is to submit the Directed Acyclic Graph process. As mentioned before, the file “SoftwareVOs.dag” which contains the lists of jobs to be processed will be submitted to the Condor Grid. The job submission is done with the command “CONDOR_SUBMIT_DAG SoftwareVOs.dag.” Once submitted we can monitor the job using the command “CONDOR_Q” which denotes the jobs that are currently in the queue. After submitting the job, the status of the Condor queue can be seen in figure 7.5:

On Figure 7.5 the Condor DAG manager has started running and its first job “Build Review” is running the Batch file “Executer.bat.” Below the Condor queue there is an updated CONDOR_STATUS command, showing the computer “Client1” in a busy state, which means that it has taken responsibility of the first job in the process.

As mentioned before, once the first job in the process is finished, there can be many parallel processes being worked on at the same time.

All of the jobs are being distributed between the computers in the pool. The requirement added in the job definitions follow which means that some jobs can only be executed by specific computers. A review of the output is explained on the next section.
7.8 Results

The job’s output is stored in individual output files. Since the same program is executed in each job, a typical output log would be as shown in the next code:

Sample output of the Build Review Process

Processing file BuildReview.txt...

File found
Printing File
Job was accepted
Y
Processing file 148.txt...

File found
The reason why the output is repeated is because there were two input parameters in the original job definition, thus the executable was ran twice.

Besides output files, there are Logfiles that store the jobs’ history. In the code below it can be seen that the job was submitted by the Broker computer (identified through its IP address, 192.168.169.131), and executed on the Client1 computer (identified through its IP address, 192.168.169.130). There is also information about the byte transmission between the two computers. There are repetitions in the bytes sent and received because the program was executed twice.

DAG Log file

000 (148.000.000) 04/12 17:35:37 Job submitted from host: <192.168.169.131:1031>

DAG Node: BuildReview

001 (148.000.000) 04/12 17:40:30 Job executing on host: <192.168.169.130:1034>

005 (148.000.000) 04/12 17:40:31 Job terminated.

(1) Normal termination (return value 0)

Usr 0 00:00:00, Sys 0 00:00:00 - Run Remote Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Run Local Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Total Remote Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Total Local Usage

771 - Run Bytes Sent By Job
438 - Run Bytes Received By Job
771 - Total Bytes Sent By Job
438 - Total Bytes Received By Job

There is a third type of log that stores errors in case something goes wrong. In this case no errors were produced, thus the error log files are empty.

The DAG file “SoftwareVOs.dag” also has the three types of Log (Log, Output, and error). This information contains technical details of the communication between the Host computer and the rest of the members of the pool. The complete listing of Log files is located in Appendix B.

7.9 Summary

The chapter describe a collaborative scenario and proceeded to implement the Condor grid to approach it. The scenario succesfully ran in the Condor Grid. The broker was able to define a Collaborative Process (through DAG) and specify the requirements for each individual step(through the job submit file). The Grid matchmaker was able to
find the proper enterprise (defined in a ClassAD) to take on the job and proceeded to send the job to it. Some jobs were able to run in parallel and the results were retrieved and documented by the Broker. As previously mentioned, real life enterprises do not work in such swift and automated manner as shown in the test run. Real scenarios take longer time to respond to job requests as well as taking longer doing the actual job. Also, the broker was able to use an automatic tool to choose the job to the proper enterprise, but the final choice is actually done by the broker with a lot analysis and planning beforehand. The fact that the broker was able to have options available with the proper documentation was enough to showcase the Grid support on VO Process definition.

The next chapter will summarize the dissertation as well as propose future work.
Chapter 8

Conclusion and Future Work

This chapter will summarize the dissertation as a whole. It will summarize the work done and describe the areas of opportunity found when working with Condor Grid.

8.1 Wrap-up

There have been other works that have promoted the need of SMES’s to integrate into VOs in order to compete with larger companies. According to [43] there hadn’t been a project that has satisfied ECOLEAD’s requirements. Most projects worked on the sole creation of a VO and a few facilitated the operation. ECOLEAD’s aim was the development of an environment where VOs could be created, operated, evolved and dissolved by designing and developing a transparent, easy to use, and affordable “plug and play” ICT infrastructure[5]. This infrastructure would play an intermediary role as enabler for interoperation among enterprises and have the support tools needed to have a smooth VO lifecycle. The Support services consist of a set of vertical application tools that would be plugged into the horizontal ICT infrastructure [9].

The ECOLEAD Project ended in 2008 and while it created many support tools to make the project a huge success, it wasn’t able to create an ICT Infrastructure that fullfilled its own proposed standards to a hundred percent[20].

The related work by Manuel Ocampo [43] proposed an architecture based on the ECOLEAD ICT-I that focused on defining business processes in a BPM language and publishing the services offered by the VO members through Web Services. The contribution focused on the definition of processes and the publication of services offered by Members of the VO which fall in the first steps of the lifecycle of the VO. This dissertation focused more on the last steps of lifecycle, orchestration and execution of the actual collaborative process.

The Condor Grid was capable of defining the whole collaborative process, it was also capable to assist the VO Broker in orchestrating the process as well as document the execution. The grid system test was able to resolve (to some extent) many interoperability issues found in the ICT infrastructure. The following is the list of issues and how the Condor Grid System approaches to resolve them:
• Access from multiple Devices: The Condor Grid is able to communicate to various computers across the internet and has support for different operating systems as well as integration to other grid technologies. The Condor grid can add neighbor pools and configure what resources can share with it. Since the grid works as middleware, it has support for higher-tier systems to be build upon it and increase its portability.

• Support for Business Collaboration: The Grid system was able to help define the whole collaborative process to be executed, each individual process within it, as well as define each enterprise available to be part of the VO. It has the power to help the broker define the requirements for the job and find the right person for it through the Matchmaker tools.

• Seamless Integration of Services and Applications: The Grid can add or change the services offered by each resource and when one becomes unavailable it can search for the next best resource within its pool as well as search in other neighboring pools of resources and notify the broker immediately.

• Heterogeneous Applications Integration: The grid supports multiple brokers as long as there are resources(or companies) available for a job. If one resource where to become unavailable, the broker can search for the next best person for the job.

• Security interoperability: The Condor system has security measures to make sure the resources are only available by the right people. It counts with quality of service features to make sure the whole process can go smoothly.

The test was able to confirm that a Grid Computing system has all the components required for a good ICT infrastructure that supports the VO cycle(specially in the VO creation and execution steps). The tool itself would require great modifications to become a full collaborative BPM system, but has shown to be able to handle the basic functions required. Condor itself showed great flexibility and simplicity to adapt to a different type of scenario that the other grid implementations could not match. It also showed great potential for improvement that will be talked about in the next section.

8.2 Future Work

It is clear that the Condor Grid has many contributions to solve the VO Integration issues, yet there are many opportunities for further research like the following:

• Adjustment of Documentation both in Enterprises as well as the Business Process.
• Creation of tools that support the full VO life Cycle. Research of other technologies that can support the VO operation

• The Condor Grid was able to support the VO process definition to a limited extent. The way that the VO process is documented as well as the way enterprises
are defined require a standard for proper documentation. This can help the broker make better decisions during the execution of the VO Process as well as make a better analysis of the performance of each enterprises for future reference.

- The prototype was only tested during the execution step of the VO lifecycle. There doesn’t exist a single toolkit that supports the full life cycle of the VO. ECOLEAD created a number of tools to support various parts of the cycle but didn’t integrate them into a toolkit or “turnkey” package.[20]

- The Condor Grid was proven as a viable technology for the support of the VO process execution but there aren’t many tools to compare it to. A research for a different technology that could act as the ICT infrastructure can be made.

8.3 Summary

The chapter described how the Condor Grid approaches each of the interoperability issues required to become the supporting ICT Infrastructure for Virtual Organization Processes. It also listed areas where the research on Grids can be explored beyond this dissertation.

Business Process Management allows enterprises to automate, monitor and improve their processes. When applied to highly dynamic environments like Virtual Organizations, BPM Suites lack the agility required. This dissertation proved how Grid Computing and the Condor Grid System in particular can work with BPM to create a better Collaborative Process tool.

The current BPM vendors are starting to focus towards Collaborative scenarios [34] and according to Gartner[27] the evolution of BPM suites will be directed towards providing BPM Software as a Service with social networking tools to support collaboration as a key to success by 2011. The Grid already shows potential to move towards that direction.
Appendices
Appendix A

Submit Files

Base.Debugging.submit

Universe = vanilla
Executable = c:\condor\bin\prueba\Executer.bat
Arguments = BaseDebugging.txt $(Cluster).txt
Requirements = Name == "Client1"
Output = c:\condor\bin\prueba\LOGS\BaseDebugging.out
Error = c:\condor\bin\prueba\LOGS\BaseDebugging.err
Log = c:\condor\bin\prueba\LOGS\BaseDebugging.log

when_to_transfer_output = ON_EXIT
getenv = true

Queue

Base.Testing.submit

Universe = vanilla
Executable = c:\condor\bin\prueba\Executer.bat
Arguments = BaseTesting.txt $(Cluster).txt
Requirements = Name == "Client1"
Output = c:\condor\bin\prueba\LOGS\BaseTesting.out
Error = c:\condor\bin\prueba\LOGS\BaseTesting.err
Log = c:\condor\bin\prueba\LOGS\BaseTesting.log

when_to_transfer_output = ON_EXIT
getenv = true

Queue

BuildApproval.Testing.submit

Universe = vanilla
Executable = c:\condor\bin\prueba\Executer.bat
Arguments = BuildApproval.txt $(Cluster).txt
Output = c:\condor\bin\prueba\LOGS\BuildApproval.out
Error = c:\condor\bin\prueba\LOGS\BuildApproval.err
Log = c:\condor\bin\prueba\LOGS\BuildApproval.log

#transfer_input_files = c:\condor\bin\archserv.exe
c:\condor\bin\archclie.exe
should_transfer_files = YES
transfer_executable = TRUE

when_to_transfer_output = ON_EXIT
getenv = true

Queue

BuildReview.Review.submit

Universe = vanilla
Executable = c:\condor\bin\prueba\Executer.bat
Arguments = BuildReview.txt $(Cluster).txt
Output = c:\condor\bin\prueba\LOGS\BuildReview.out
Error = c:\condor\bin\prueba\LOGS\BuildReview.err
Log = c:\condor\bin\prueba\LOGS\BuildReview.log

when_to_transfer_output = ON_EXIT
getenv = true

Queue

Calc.Debugging.submit

Universe = vanilla
Executable = c:\condor\bin\prueba\Executer.bat
Arguments = CalcDebugging.txt $(Cluster).txt
Requirements = Name == "Client2"
Output = c:\condor\bin\prueba\LOGS\CalcDebugging.out
Error = c:\condor\bin\prueba\LOGS\CalcDebugging.err
Log = c:\condor\bin\prueba\LOGS\CalcDebugging.log

when_to_transfer_output = ON_EXIT
getenv = true

Queue

Calc.Testing.submit

Universe = vanilla
Executable = c:\condor\bin\prueba\Executer.bat
Arguments = CalcTesting.txt $(Cluster).txt
Requirements = Name == "Client2"
Output = c:\condor\bin\prueba\LOGS\CalcTesting.out
Error = c:\condor\bin\prueba\LOGS\CalcTesting.err
Log = c:\condor\bin\prueba\LOGS\CalcTesting.log

when_to_transfer_output = ON_EXIT
getenv = true

Queue

Draw.Debugging.submit

Universe = vanilla
Executable = c:\condor\bin\prueba\Executer.bat
Arguments = DrawDebugging.txt $(Cluster).txt
Requirements = Name="client2"
Output = c:\condor\bin\prueba\LOGS\DrawDebugging.out
Error = c:\condor\bin\prueba\LOGS\DrawDebugging.err
Log = c:\condor\bin\prueba\LOGS\DrawDebugging.log

when_to_transfer_output = ON_EXIT
getenv = true

Queue

Draw.Testing.submit

Universe = vanilla
Executable = c:\condor\bin\prueba\Executer.bat
Arguments = DrawTesting.txt $(Cluster).txt
Requirements = Name == "Client2"
Output = c:\condor\bin\prueba\LOGS\DrawTesting.out
Error = c:\condor\bin\prueba\LOGS\DrawTesting.err
Log = c:\condor\bin\prueba\LOGS\DrawTesting.log

when_to_transfer_output = ON_EXIT
getenv = true

Queue

Impress.Debugging.submit

Universe = vanilla
Executable = c:\condor\bin\prueba\Executer.bat
Arguments = ImpressDebugging.txt $(Cluster).txt
Output = c:\condor\bin\prueba\LOGS\ImpressDebugging.out
Error = c:\condor\bin\prueba\LOGS\ImpressDebugging.err
Log = c:\condor\bin\prueba\LOGS\ImpressDebugging.log

when_to_transfer_output = ON_EXIT
getenv = true

Queue

Impress.Testing.submit

Universe  = vanilla
Executable = c:\condor\bin\prueba\Executer.bat
Arguments  = ImpressTesting.txt $(Cluster).txt
Requirements = Name == "Client1"
Output = c:\condor\bin\prueba\LOGS\ImpressTesting.out
Error = c:\condor\bin\prueba\LOGS\ImpressTesting.err
Log = c:\condor\bin\prueba\LOGS\ImpressTesting.log

when_to_transfer_output = ON_EXIT
getenv = true

Queue

Writer.Debugging.submit

Universe  = vanilla
Executable = c:\condor\bin\prueba\Executer.bat
Arguments  = WriterDebugging.txt $(Cluster).txt
Requirements = Name == "Client2"
Output = c:\condor\bin\prueba\LOGS\WriterDebugging.out
Error = c:\condor\bin\prueba\LOGS\WriterDebugging.err
Log = c:\condor\bin\prueba\LOGS\WriterDebugging.log

when_to_transfer_output = ON_EXIT
getenv = true

Queue

Writer.Testing.submit

Universe  = vanilla
Executable = c:\condor\bin\prueba\Executer.bat
Arguments  = WriterTesting.txt $(Cluster).txt
Requirements = Name == "Client1"
Output = c:\condor\bin\prueba\LOGS\WriterTesting.out
Error = c:\condor\bin\prueba\LOGS\WriterTesting.err
Log = c:\condor\bin\prueba\LOGS\WriterTesting.log

when_to_transfer_output = ON_EXIT
getenv = true

Queue
Appendix B

Dag Files

SoftwareVOs.dag.dagman.out

4/12 17:35:22 ****************************************************
4/12 17:35:22 ** condor_scheduniv_exec.147.0 (CONDOR_DAGMAN)
4/12 17:35:22 ** STARTING UP
4/12 17:35:22 ** C:\condor\bin\condor_dagman.exe
4/12 17:35:22 ** $CondorVersion: 7.0.5 Sep 20 2008 BuildID: 105846 $
4/12 17:35:22 ** $CondorPlatform: INTEL-WINNT50 $
4/12 17:35:22 ** PID = 1100
4/12 17:35:22 ** Log last touched time unavailable
(No such file or directory)
4/12 17:35:22 ****************************************************
4/12 17:35:22 Using config source: C:\condor\condor_config
4/12 17:35:22 Using local config sources:
4/12 17:35:22 C:\condor/condor_config.local
4/12 17:35:23 DaemonCore: Command Socket at <192.168.169.131:2101>
4/12 17:35:23 DAGMAN_SUBMIT_DELAY setting: 0
4/12 17:35:23 DAGMAN_MAX_SUBMIT_ATTEMPTS setting: 6
4/12 17:35:23 DAGMAN_STARTUP_CYCLE_DETECT setting: 0
4/12 17:35:23 DAGMAN_MAX_SUBMITS_PER_INTERVAL setting: 5
4/12 17:35:23 allow_events (DAGMAN_IGNORE_DUPLICATE_JOB_EXECUTION, DAGMAN_ALLOW_EVENTS) setting: 114
4/12 17:35:23 DAGMAN_RETRY_SUBMIT_FIRST setting: 1
4/12 17:35:23 DAGMAN_RETRY_NODE_FIRST setting: 0
4/12 17:35:23 DAGMAN_MAX_JOBS_IDLE setting: 0
4/12 17:35:23 DAGMAN_MAX_JOBS_SUBMITTED setting: 0
4/12 17:35:23 DAGMAN_MUNGGE_NODE_NAMES setting: 1
4/12 17:35:23 DAGMAN_DELETE_OLD_LOGS setting: 1
4/12 17:35:23 DAGMAN_PROHIBIT_MULTI_JOBS setting: 0
4/12 17:35:23 DAGMAN_SUBMIT_DEPTH_FIRST setting: 0
4/12 17:35:23 DAGMAN_ABORT_DUPLICATES setting: 1
4/12 17:35:23 DAGMAN_ABORT_ON_SCARY_SUBMIT setting: 1
4/12 17:35:23 DAGMAN_PENDING_REPORT_INTERVAL setting: 600
4/12 17:35:23 argv[0] == "condor_scheduniv_exec.147.0"
4/12 17:35:23 argv[1] == "-Debug"
4/12 17:35:23 argv[2] == "3"
4/12 17:35:23 argv[3] == "-Lockfile"
4/12 17:35:23 argv[4] == "SoftwareVOs.dag.lock"
4/12 17:35:23 argv[5] == "-Condorlog"
4/12 17:35:23 argv[6] == "c:\condor\bin\prueba\LOGS\BuildReview.log"
4/12 17:35:23 argv[7] == "-Dag"
4/12 17:35:23 argv[8] == "SoftwareVOs.dag"
4/12 17:35:23 argv[9] == "-Rescue"
4/12 17:35:23 argv[10] == "SoftwareVOs.dag.rescue"
4/12 17:35:23 DAG Lockfile will be written to SoftwareVOs.dag.lock
4/12 17:35:23 DAG Input file is SoftwareVOs.dag
4/12 17:35:23 Rescue DAG will be written to SoftwareVOs.dag.rescue
4/12 17:35:23 All DAG node user log files:
4/12 17:35:23 c:\condor\bin\prueba\LOGS\BuildReview.log
4/12 17:35:23 c:\condor\bin\prueba\LOGS\WriterDebugging.log
4/12 17:35:23 c:\condor\bin\prueba\LOGS\CalcDebugging.log
4/12 17:35:23 c:\condor\bin\prueba\LOGS\ImpressDebugging.log
4/12 17:35:23 c:\condor\bin\prueba\LOGS\DrawDebugging.log
4/12 17:35:23 c:\condor\bin\prueba\LOGS\WriterTesting.log
4/12 17:35:23 c:\condor\bin\prueba\LOGS\CalcTesting.log
4/12 17:35:23 c:\condor\bin\prueba\LOGS\ImpressTesting.log
4/12 17:35:23 c:\condor\bin\prueba\LOGS\DrawTesting.log
4/12 17:35:23 c:\condor\bin\prueba\LOGS\BuildApproval.log
4/12 17:35:23 Parsing SoftwareVOs.dag ...
4/12 17:35:23 Dag contains 10 total jobs
4/12 17:35:23 Truncating any older versions of log files...
4/12 17:35:23 Sleeping for 12 seconds to ensure ProcessId uniqueness
4/12 17:35:35 WARNING: ProcessId not confirmed unique
4/12 17:35:35 Bootstrapping...
4/12 17:35:35 Number of pre-completed nodes: 0
4/12 17:35:35 Registering condor_event_timer...
4/12 17:35:36 Submitting Condor Node BuildReview job(s)...
4/12 17:35:36 submitting: condor_submit -a dag_node_name'
  '-' 'BuildReview -a +DAGManJobId' '-=' '147 -a DAGManJobId'
  '-=' '147 -a submit_event_notes' '-=' 'DAG' 'Node:'
  'BuildReview -a +DAGParentNodeNames' '-=
  "" BuildReview.Review.submit
4/12 17:35:37 From submit: Submitting job(s).
4/12 17:35:37 From submit: Logging submit event(s).
4/12 17:35:37 From submit: 1 job(s) submitted to cluster 148.
4/12 17:35:37 assigned Condor ID (148.0)
4/12 17:35:37 Just submitted 1 job this cycle...
4/12 17:35:38 Event: ULOG_SUBMIT for Condor Node BuildReview (148.0)
4/12 17:35:38 Number of idle job procs: 1
4/12 17:35:38 == Done Pre Queued Post Ready Un-Ready Failed ==
4/12 17:35:38 0 0 1 0 0 9 0
4/12 17:40:33 Event: ULOG_EXECUTE for Condor Node BuildReview (148.0)
4/12 17:40:33 Number of idle job procs: 0
4/12 17:40:33 == Done Pre Queued Post Ready Un-Ready Failed ==
4/12 17:40:33 1 0 0 0 4 5 0
4/12 17:40:38 Sleeping for one second for log file consistency
4/12 17:40:39 Submitting Condor Node WriterMod job(s)...
4/12 17:40:39 submitting: condor_submit -a dag_node_name'
  '-' 'WriterMod -a +DAGManJobId' '-=' '147 -a DAGManJobId'
  '-=' '147 -a submit_event_notes' '-=' 'DAG' 'Node:'
  'WriterMod -a +DAGParentNodeNames' '-=
  "" BuildReview.Writer.Debugging.submit
4/12 17:40:40 From submit: Submitting job(s).
4/12 17:40:40 From submit: Logging submit event(s).
4/12 17:40:40 From submit: 1 job(s) submitted to cluster 149.
4/12 17:40:40  assigned Condor ID (149.0)
4/12 17:40:40  Sleeping for one second for
log file consistency
4/12 17:40:41  Submitting Condor Node CalcMod job(s)...
4/12 17:40:41  submitting: condor_submit -a dag_node_name'
   '=' 'CalcMod -a +DAGManJobId' '=' '147 -a DAGManJobId'
   '=' '147 -a submit_event_notes' '=' 'DAG' 'Node:'
   'CalcMod -a +DAGParentNodeNames' '=' '"BuildReview"'
Calc.Debugging.submit
4/12 17:40:42  From submit: Submitting job(s).
4/12 17:40:42  From submit: Logging submit event(s).
4/12 17:40:42  From submit: 1 job(s) submitted
to cluster 150.
4/12 17:40:42  assigned Condor ID (150.0)
4/12 17:40:42  Sleeping for one second for
log file consistency
4/12 17:40:43  Submitting Condor Node
ImpressMod job(s)...
4/12 17:40:43  submitting: condor_submit -a
dag_node_name' '=' 'ImpressMod -a +DAGManJobId'
   '=' '147 -a DAGManJobId' '=' '147 -a
submit_event_notes' '=' 'DAG' 'Node:'
   'ImpressMod -a +DAGParentNodeNames'
   '=' '"BuildReview" Impress.Debugging.submit
4/12 17:40:44  From submit: Submitting job(s).
4/12 17:40:44  From submit: Logging submit event(s).
4/12 17:40:44  From submit: 1 job(s) submitted
to cluster 151.
4/12 17:40:44  assigned Condor ID (151.0)
4/12 17:40:44  Sleeping for one second for
log file consistency
4/12 17:40:45  Submitting Condor Node
DrawMod job(s)...
4/12 17:40:45  submitting: condor_submit
   -a dag_node_name'
   '=' 'DrawMod -a +DAGManJobId' '=' '147
-a DAGManJobId'
   '=' '147 -a submit_event_notes' '=' 'DAG'
   'Node: ' 'DrawMod -a +DAGParentNodeNames'
   '=' '"BuildReview" Draw.Debugging.submit
4/12 17:40:47  From submit: Submitting job(s).
4/12 17:40:47  From submit: Logging submit
event(s).
4/12 17:40:47  From submit: 1 job(s)
submitted to cluster 152.
4/12 17:40:47  assigned Condor ID (152.0)
4/12 17:40:47 Just submitted 4 jobs this cycle...
4/12 17:40:47 Event: ULOG_SUBMIT for Condor Node WriterMod (149.0)
4/12 17:40:47 Number of idle job procs: 1
4/12 17:40:47 Event: ULOG_SUBMIT for Condor Node CalcMod (150.0)
4/12 17:40:47 Number of idle job procs: 2
4/12 17:40:47 Event: ULOG_SUBMIT for Condor Node ImpressMod (151.0)
4/12 17:40:47 Number of idle job procs: 3
4/12 17:40:47 Event: ULOG_SUBMIT for Condor Node DrawMod (152.0)
4/12 17:40:47 Number of idle job procs: 4
4/12 17:40:47 Of 10 nodes total:
4/12 17:40:47 Done Pre Queued Post Ready Un-Ready Failed
4/12 17:40:47 === === === === === === ===
4/12 17:40:47 1 0 4 0 0 5 0
4/12 17:40:52 Event: ULOG_EXECUTE for Condor Node WriterMod (149.0)
4/12 17:40:52 Number of idle job procs: 3
4/12 17:40:52 Event: ULOG_JOB_TERMINATED for Condor Node WriterMod (149.0)
4/12 17:40:52 Node WriterMod job proc (149.0) completed successfully.
4/12 17:40:52 Node WriterMod job completed
4/12 17:40:52 Number of idle job procs: 3
4/12 17:40:52 Event: ULOG_EXECUTE for Condor Node CalcMod (150.0)
4/12 17:40:52 Number of idle job procs: 2
4/12 17:40:52 Event: ULOG_JOB_TERMINATED for Condor Node CalcMod (150.0)
4/12 17:40:52 Node CalcMod job proc (150.0) completed successfully.
4/12 17:40:52 Node CalcMod job completed
4/12 17:40:52 Number of idle job procs: 2
4/12 17:40:52 Event: ULOG_EXECUTE for Condor Node ImpressMod (151.0)
4/12 17:40:52 Number of idle job procs: 1
4/12 17:40:52 Of 10 nodes total:
4/12 17:40:52 Done Pre Queued Post Ready Un-Ready Failed
4/12 17:40:52 === === === === === === ===
4/12 17:40:52 3 0 2 0 2 3 0
4/12 17:40:57 Sleeping for one second for log file consistency
4/12 17:40:58 Submitting Condor Node
WriterTest job(s)...
4/12 17:40:58 submitting: condor_submit
-a dag_node_name'
'=' 'WriterTest -a +DAGManJobId' '='
'147 -a DAGManJobId' '=' '147 -a
submit_event_notes' '=' 'DAG' 'Node:'
'WriterTest -a +DAGParentNodeNames' '='
"WriterMod" Writer.Testing.submit
4/12 17:40:59 From submit: Submitting
job(s).
4/12 17:40:59 From submit: Logging
submit event(s).
4/12 17:40:59 From submit: 1 job(s)
submitted to cluster 153.
4/12 17:40:59 assigned Condor ID (153.0)
4/12 17:40:59 Sleeping for one second
for log file consistency
4/12 17:41:00 Submitting Condor Node
CalcTest job(s)...
4/12 17:41:00 submitting: condor_submit
-a dag_node_name'
'=' 'CalcTest -a +DAGManJobId' '='
'147 -a DAGManJobId' '=' '147 -a
submit_event_notes' '=' 'DAG' 'Node:'
'CalcTest -a +DAGParentNodeNames' '='
"CalcMod" Calc.Testing.submit
4/12 17:41:01 From submit: Submitting
job(s).
4/12 17:41:01 From submit: Logging
submit event(s).
4/12 17:41:01 From submit: 1 job(s)
submitted to cluster 154.
4/12 17:41:01 assigned Condor ID (154.0)
4/12 17:41:01 Just submitted 2 jobs
this cycle...
4/12 17:41:01 Event: ULOG_JOB_TERMINATED
for Condor
Node ImpressMod (151.0)
4/12 17:41:01 Node ImpressMod job
proc (151.0) completed successfully.
4/12 17:41:01 Node ImpressMod job completed
4/12 17:41:01 Number of idle job procs: 1
4/12 17:41:01 Event: ULOG_EXECUTE for Condor
Node DrawMod (152.0)
4/12 17:41:01 Number of idle job procs: 0
4/12 17:41:01 Event: ULOG_JOB_TERMINATED

82
for Condor Node DrawMod (152.0)
4/12 17:41:01 Node DrawMod job proc (152.0) completed successfully.
4/12 17:41:01 Node DrawMod job completed
4/12 17:41:01 Number of idle job procs: 0
4/12 17:41:01 Event: ULOG_SUBMIT for Condor Node WriterTest (153.0)
4/12 17:41:01 Number of idle job procs: 1
4/12 17:41:01 Event: ULOG_SUBMIT for Condor Node CalcTest (154.0)
4/12 17:41:01 Number of idle job procs: 2
4/12 17:41:01 Of 10 nodes total:
4/12 17:41:01 Done Pre Queued Post Ready Un-Ready Failed
4/12 17:41:01 === === === === === === ===
4/12 17:41:01 5 0 2 0 2 1 0
4/12 17:41:06 Sleeping for one second for log file consistency
4/12 17:41:07 Submitting Condor Node ImpressTest job(s)...
4/12 17:41:07 submitting: condor_submit
-a dag_node_name' '=' 'ImpressTest -a
+DAGManJobId' '=' '147 -a DAGManJobId'
'=' '147 -a submit_event_notes' '='
'DAG' 'Node:' 'ImpressTest -a
+DAGParentNodeNames' '=' '"ImpressMod" Impress.Testing.submit
4/12 17:41:09 From submit: Submitting job(s).
4/12 17:41:09 From submit: Logging submit event(s).
4/12 17:41:09 From submit: 1 job(s) submitted to cluster 155.
4/12 17:41:09 assigned Condor ID (155.0)
4/12 17:41:09 Sleeping for one second for log file consistency
4/12 17:41:10 Submitting Condor Node DrawTest job(s)...
4/12 17:41:10 submitting: condor_submit
-a dag_node_name'
'=' 'DrawTest -a +DAGManJobId' '='
'147 -a DAGManJobId'
'=' '147 -a submit_event_notes' '='
'DAG' 'Node:' 'DrawTest -a
+DAGParentNodeNames' '='
"DrawMod" Draw.Testing.submit
4/12 17:41:12 From submit: Submitting
job(s).
4/12 17:41:12 From submit: Logging submit event(s).
4/12 17:41:12 From submit: 1 job(s) submitted to cluster 156.
4/12 17:41:12 assigned Condor ID (156.0)
4/12 17:41:12 Just submitted 2 jobs to this cycle...
4/12 17:41:12 Event: ULOG_EXECUTE for Condor Node CalcTest (154.0)
4/12 17:41:12 Number of idle job procs: 1
4/12 17:41:12 Event: ULOG_JOB_TERMINATED for Condor Node CalcTest (154.0)
4/12 17:41:12 Node CalcTest job proc (154.0) completed successfully.
4/12 17:41:12 Event: ULOG_SUBMIT for Condor Node ImpressTest (155.0)
4/12 17:41:12 Event: ULOG_SUBMIT for Condor Node DrawTest (156.0)
4/12 17:41:12 Number of idle job procs: 3
4/12 17:41:12 Of 10 nodes total:

```
4/12 17:41:12 Done Pre Queued Post Ready Un-Ready Failed
4/12 17:41:12 = = = = = = = = = =
4/12 17:41:12 6 0 3 0 0 1 0
```

4/12 17:41:12 Event: ULOG_EXECUTE for Condor Node WriterTest (153.0)
4/12 17:41:12 Number of idle job procs: 2
4/12 17:41:22 Event: ULOG_JOB_TERMINATED for Condor Node WriterTest (153.0)
4/12 17:41:22 Node WriterTest job proc (153.0) completed successfully.
4/12 17:41:22 Event: ULOG_SUBMIT for Condor Node ImpressTest (155.0)
4/12 17:41:22 Number of idle job procs: 1
4/12 17:41:22 Of 10 nodes total:

```
4/12 17:41:22 Done Pre Queued Post Ready Un-Ready Failed
4/12 17:41:22 = = = = = = = = = =
4/12 17:41:22 7 0 2 0 0 1 0
```

4/12 17:41:27 Event: ULOG_EXECUTE for Condor Node ImpressTest (155.0)
4/12 17:41:27 Number of idle job procs: 1
4/12 17:41:27 Event: ULOG_JOB_TERMINATED for Condor Node ImpressTest (155.0)
4/12 17:41:27 Node ImpressTest job proc (155.0) completed successfully.
4/12 17:41:27 Node ImpressTest job completed
4/12 17:41:27 Number of idle job procs: 1
4/12 17:41:27 Of 10 nodes total:
4/12 17:41:27 Done Pre Queued Post Ready Un-Ready Failed
4/12 17:41:27 === === === === === === === ===
4/12 17:41:27 8 0 1 0 0 1 0
4/12 17:41:32 Event: ULOG_EXECUTE for Condor Node DrawTest (156.0)
4/12 17:41:32 Number of idle job procs: 0
4/12 17:41:32 Event: ULOG_JOB_TERMINATED for Condor Node DrawTest (156.0)
4/12 17:41:32 Node DrawTest job proc (156.0) completed successfully.
4/12 17:41:32 Node DrawTest job completed
4/12 17:41:32 Number of idle job procs: 0
4/12 17:41:32 Of 10 nodes total:
4/12 17:41:32 Done Pre Queued Post Ready Un-Ready Failed
4/12 17:41:32 === === === === === === === ===
4/12 17:41:32 9 0 0 0 1 0 0
4/12 17:41:37 Sleeping for one second for log file consistency
4/12 17:41:38 Submitting Condor Node BuildApproval job(s)...
4/12 17:41:38 submitting: condor_submit -a
dag_node_name' ' '=' 'BuildApproval -a
+DAGManJobId' ' '=' '147 -a DAGManJobId'
'=' '147 -a submit_event_notes' ' '='
'DAG' 'Node' 'BuildApproval -a
+DAGParentNodeNames' ' '='
"WriterTest,CalcTest,ImpressTest,DrawTest"
BuildApproval.Testing.submit
4/12 17:41:39 From submit: Submitting job(s).
4/12 17:41:39 From submit: Logging submit event(s).
4/12 17:41:39 From submit: 1 job(s) submitted to cluster 157.
4/12 17:41:39 assigned Condor ID (157.0)
4/12 17:41:39 Just submitted 1 job this cycle...
4/12 17:41:39 Event: ULOG_SUBMIT for Condor Node
BuildApproval (157.0)
4/12 17:41:39 Number of idle job procs: 1
4/12 17:41:39 Of 10 nodes total:
4/12 17:41:39 Done Pre Queued Post Ready Un-Ready Failed
4/12 17:41:39 === === === === === === ===
4/12 17:41:39 9 0 1 0 0 0 0
4/12 17:41:49 Event: ULOG_EXECUTE for Condor Node BuildApproval (157.0)
4/12 17:41:49 Number of idle job procs: 0
4/12 17:41:49 Event: ULOG_JOB_TERMINATED for Condor Node BuildApproval (157.0)
4/12 17:41:49 Node BuildApproval job proc (157.0) completed successfully.
4/12 17:41:49 Node BuildApproval job completed
4/12 17:41:49 Number of idle job procs: 0
4/12 17:41:49 Of 10 nodes total:
4/12 17:41:49 Done Pre Queued Post Ready Un-Ready Failed
4/12 17:41:49 === === === === === === ===
4/12 17:41:49 10 0 0 0 0 0 0
4/12 17:41:49 All jobs Completed!
4/12 17:41:49 Note: 0 total job deferrals because of -MaxJobs limit (0)
4/12 17:41:49 Note: 0 total job deferrals because of -MaxIdle limit (0)
4/12 17:41:49 Note: 0 total job deferrals because of node category throttles
4/12 17:41:49 Note: 0 total PRE script deferrals because of -MaxPre limit (0)
4/12 17:41:49 Note: 0 total POST script deferrals because of -MaxPost limit (0)
4/12 17:41:49 **** condor_scheduniv_exec.147.0 (condor_DAGMAN) EXITING WITH STATUS 0

SoftwareVOs.dag

JOB BuildReview BuildReview.Review.submit
JOB WriterMod Writer.Debugging.submit
JOB CalcMod Calc.Debugging.submit
JOB ImpressMod Impress.Debugging.submit
JOB DrawMod Draw.Debugging.submit
JOB WriterTest Writer.Testing.submit
JOB CalcTest Calc.Testing.submit
JOB ImpressTest Impress.Testing.submit
JOB DrawTest Draw.Testing.submit
JOB BuildApproval BuildApproval.Testing.submit

PARENT BuildReview CHILD WriterMod CalcMod ImpressMod DrawMod
SoftwareVOs.dag.dagman.log

000 (147.000.000) 04/12 17:35:21 Job submitted from host:
<192.168.169.131:1031>
...

001 (147.000.000) 04/12 17:35:22 Job executing on host:
<192.168.169.131:1031>
...

005 (147.000.000) 04/12 17:41:49 Job terminated.
(1) Normal termination (return value 0)
Usr 0 00:00:00, Sys 0 00:00:00 - Run Remote Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Run Local Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Total Remote Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Total Local Usage
0 - Run Bytes Sent By Job
0 - Run Bytes Received By Job
0 - Total Bytes Sent By Job
0 - Total Bytes Received By Job
...
Appendix C

Source Code

Executer.bat

```bash
@echo off
if "%1" == "" goto error
rem - process each of the named files
:again
rem if %1 is blank, we are finished
if "%1" == "" goto end
echo.
echo Processing file %1...
type %1
archserv.exe %1 %2 peticion broker
rem - shift the arguments and examine %1 again
shift
:again
:error
echo missing argument!
echo usage   view file1.txt view2.txt ...
:end
echo.
echo Done.
```

FILEA.TXT

```
C:\CONDOR\BIN\PRUEBA\ARCHSERV.EXE
C:\condor\bin\prueba\Writer.Debugging.submit
peticion
broker
```

ARCHSERV.CPP
```cpp
#include <fstream.h>
#include <dos.h>
#include <iostream.h>
#include <conio.h>

//The program creates an file in the receiving computer
//The data inside the file is taken from the arguments
//Once writen, the program looks for another file with
//a reply
//The program read the reply then ends.

int main(int argc, char* argv[]) {
    clrscr();
    ifstream Test2;
    ofstream out;
    out.open("filea.txt", ifstream::trunc);
    for (int i=0; i<argc; i++)
        out<<argv[i]<<endl;
    out.close();
    Test2.open("fileb.txt", ifstream::in);
    Test2.close();
    while (Test2.fail())
    {
        cout<<"Standby for reply\n";
        sleep(5);
        Test2.open("fileb.txt", ifstream::in);
        Test2.close();
        clrscr();
    }
    cout<<"\nFile found\n";
    Test2.open("fileb.txt", ifstream::in);
    char tem;
    cout<<"imprimiendo archivo \n";
    int accept=1;
    while(Test2.get(tem))
    {
        if (tem=='y' || tem=='Y')
            cout<<"Job was accepted \n";
        else if (tem=='n' || tem=='N')
        {
            cout<<"Job was denied\n";
            accept=0;
            break;
        }
    }
}
```

ARCHCLIE.CPP

#include <fstream.h>
#include <dos.h>
#include <iostream.h>
#include <conio.h>

// The program checks for the creation of a specific file
// Once detected, the information will be inputed and
// replied with a yes
// A new file will be created to reply.

int main(int argc, char* argv[]) {  
    char* FileName;  
    if (argc>0)  
        FileName = argv[1];
    clrscr();  
    ifstream Test2;  
    ofstream out;
    // Change filea to arguments  
    Test2.open("filea.txt", ifstream::in);  
    Test2.close();  
    while (Test2.fail())  
    {  
        cout<<"Standby por peticiones\n";  
        sleep(5);  
        Test2.open ("filea.txt", ifstream::in);  
        Test2.close();  
        clrscr();  
    }  
    cout<<"\n File found \n";  
    Test2.open ("filea.txt", ifstream::in);  
    char tem;  
    cout<<"Printing file \n";  
    while(Test2.get(tem))  
    {  
        cout<<tem;  
    }  
    cout<<"\n Accept JOb? Y/N \n ";
}
cin>>tem;
if (tem==’y’ || tem==’Y’)
{
    out.open ("fileb.txt", ios::trunc);
    out<<"Y";
}
else
{
    out.open ("fileb.txt", ios::trunc);
    out<<"N";
}
cout<<argc<<endl<<FileName;
return 0;
Appendix D

Out Files

BuildApproval.out

Volume in drive C has no label.
Volume Serial Number is 1C2F-0294

Directory of C:\condor\execute\dir_1480

04/12/2009 05:42 PM <DIR> .
04/12/2009 05:42 PM <DIR> ..
04/12/2009 05:39 PM 438 condor_exec.bat
04/12/2009 05:42 PM 0 _condor_stderr
04/12/2009 05:42 PM 0 _condor_stdout
3 File(s) 438 bytes
2 Dir(s) 6,350,569,472 bytes free

Processing file BuildApproval.txt...
File found
Printing file
Job was accepted
Y
Processing file 157.txt...
File found
Printing file
Job was accepted
Y
Done.

BuildReview.out

Volume in drive C has no label.
Volume Serial Number is 1C2F-0294
Directory of C:\condor\execute\dir_1112

04/12/2009  05:40 PM  <DIR> .
04/12/2009  05:40 PM  <DIR> ..
04/12/2009  05:37 PM  438 condor_exec.bat
04/12/2009  05:40 PM  0 _condor_stderr
04/12/2009  05:40 PM  0 _condor_stdout

3 File(s) 438 bytes
2 Dir(s) 6,350,577,664 bytes free

Processing file BuildReview.txt...

File found
Printing File
Job was accepted
Y
Processing file 148.txt...

File found
Printing file
Job was accepted
Y
Done.

CalcDebugging.out

Volume in drive C has no label.
Volume Serial Number is 1C2F-0294

Directory of C:\condor\execute\dir_1228

04/12/2009  05:41 PM  <DIR> .
04/12/2009  05:41 PM  <DIR> ..
04/12/2009  05:38 PM  438 condor_exec.bat
04/12/2009  05:41 PM  0 _condor_stderr
04/12/2009  05:41 PM  0 _condor_stdout

3 File(s) 438 bytes
2 Dir(s) 6,350,585,856 bytes free

Processing file CalcDebugging.txt...

File found
Printing file
Job was accepted
Y
Processing file 150.txt...

File found
Printing file
Job was accepted
Y
Done.

CalcTesting.out

Volume in drive C has no label.
Volume Serial Number is 1C2F-0294

Directory of C:\condor\execute\dir_1696

04/12/2009  05:41 PM    <DIR>  .
04/12/2009  05:41 PM    <DIR>  ..
04/12/2009  05:38 PM    438  condor_exec.bat
04/12/2009  05:41 PM    0  _condor_stderr
04/12/2009  05:41 PM    0  _condor_stdout

  3 File(s)     438 bytes
  2 Dir(s) 6,350,573,568 bytes free

Processing file CalcTesting.txt...

File found
Processing file
Job was accepted
Y
Processing file 154.txt...

File found
Processing file
Job was accepted
Y
Done.

DrawDebugging.out

Volume in drive C has no label.
Volume Serial Number is 1C2F-0294

Directory of C:\condor\execute\dir_280

04/12/2009  05:41 PM    <DIR>  .
04/12/2009  05:41 PM    <DIR> ..
04/12/2009  05:38 PM    438  condor_exec.bat
04/12/2009  05:41 PM    0  _condor_stderr
04/12/2009  05:41 PM    0  _condor_stdout

  3 File(s)     438 bytes
  2 Dir(s) 6,350,581,760 bytes free
Processing file DrawDebugging.txt...

File found
Printing file
Job was accepted
Y
Processing file 152.txt...

File found
Printing file
Job was accepted
Y
Done.

DrawTesting.out

Volume in drive C has no label.
Volume Serial Number is 1C2F-0294

Directory of C:\condor\execute\dir_1200

04/12/2009  05:38 PM  condor_exec.bat
04/12/2009  05:41 PM  _condor_stderr
04/12/2009  05:41 PM  _condor_stdout

3 File(s)  438 bytes
2 Dir(s)  6,350,573,568 bytes free

Processing file DrawTesting.txt...

File found
Printing file
Job was accepted
Y
Processing file 156.txt...

File found
Printing file
Job was accepted
Y
Done.

ImpressDebugging.out

Volume in drive C has no label.
Volume Serial Number is 1C2F-0294
Directory of C:\condor\execute\dir_892

04/12/2009 05:41 PM  <DIR>     .
04/12/2009 05:41 PM  <DIR>     ..
04/12/2009 05:38 PM  438 condor_exec.bat
04/12/2009 05:41 PM  0 _condor_stderr
04/12/2009 05:41 PM  0 _condor_stdout

3 File(s) 438 bytes
2 Dir(s) 6,350,581,760 bytes free

Processing file ImpressDebugging.txt...

File found
Printing file
Job was accepted
Y
Processing file 151.txt...

File found
Printing file
Job was accepted
Y
Done.

ImpressTesting.out

Volume in drive C has no label.
Volume Serial Number is 1C2F-0294

Directory of C:\condor\execute\dir_1180

04/12/2009 05:41 PM  <DIR>     .
04/12/2009 05:41 PM  <DIR>     ..
04/12/2009 05:38 PM  438 condor_exec.bat
04/12/2009 05:41 PM  0 _condor_stderr
04/12/2009 05:41 PM  0 _condor_stdout

3 File(s) 438 bytes
2 Dir(s) 6,341,169,152 bytes free

Processing file ImpressTesting.txt...

File found
Printing file
Job was accepted
Y
Processing file 155.txt...
File found
Printing file
Job was accepted
Y
Done.

**WriterDebugging.out**

Volume in drive C has no label.
Volume Serial Number is 1C2F-0294

Directory of C:\condor\execute\dir_204

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/12/2009</td>
<td>05:41 PM</td>
<td>&lt;DIR&gt;</td>
<td>.</td>
</tr>
<tr>
<td>04/12/2009</td>
<td>05:41 PM</td>
<td>&lt;DIR&gt;</td>
<td>..</td>
</tr>
<tr>
<td>04/12/2009</td>
<td>05:38 PM</td>
<td></td>
<td>condor_exec.bat</td>
</tr>
<tr>
<td>04/12/2009</td>
<td>05:41 PM</td>
<td></td>
<td>_condor_stderr</td>
</tr>
<tr>
<td>04/12/2009</td>
<td>05:41 PM</td>
<td></td>
<td>_condor_stdout</td>
</tr>
</tbody>
</table>

3 File(s) 438 bytes
2 Dir(s) 6,350,585,856 bytes free

Processing file WriterDebugging.txt...

File found
Printing file
Job was accepted
Y
Processing file 149.txt...

File found
Printing file
Job was accepted
Y
Done.

**WriterTesting.out**

Volume in drive C has no label.
Volume Serial Number is 1C2F-0294

Directory of C:\condor\execute\dir_1088

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/12/2009</td>
<td>05:41 PM</td>
<td>&lt;DIR&gt;</td>
<td>.</td>
</tr>
<tr>
<td>04/12/2009</td>
<td>05:41 PM</td>
<td>&lt;DIR&gt;</td>
<td>..</td>
</tr>
<tr>
<td>04/12/2009</td>
<td>05:38 PM</td>
<td></td>
<td>condor_exec.bat</td>
</tr>
<tr>
<td>04/12/2009</td>
<td>05:41 PM</td>
<td></td>
<td>_condor_stderr</td>
</tr>
<tr>
<td>04/12/2009</td>
<td>05:41 PM</td>
<td></td>
<td>_condor_stdout</td>
</tr>
</tbody>
</table>

3 File(s) 438 bytes
2 Dir(s)  6,341,181,440 bytes free

Processing file WriterTesting.txt...

   File found
   Printing file
   Job was accepted
   Y
Processing file 153.txt...

   File found
   Printing file
   Job was accepted
   Y
Done.
Appendix E

Log Files

BuildApproval.log

000 (157.000.000) 04/12 17:41:39 Job submitted from host:
<192.168.169.131:1031>
   DAG Node: BuildApproval
   ...
001 (157.000.000) 04/12 17:41:48 Job executing on host:
<192.168.169.130:1034>
   ...
005 (157.000.000) 04/12 17:41:49 Job terminated.
   (1) Normal termination (return value 0)
   Usr 0 00:00:00, Sys 0 00:00:00 - Run Remote Usage
   Usr 0 00:00:00, Sys 0 00:00:00 - Run Local Usage
   Usr 0 00:00:00, Sys 0 00:00:00 - Total Remote Usage
   Usr 0 00:00:00, Sys 0 00:00:00 - Total Local Usage
   773 - Run Bytes Sent By Job
   438 - Run Bytes Received By Job
   773 - Total Bytes Sent By Job
   438 - Total Bytes Received By Job
   ...

BuildReview.log

000 (148.000.000) 04/12 17:35:37 Job submitted from host:
<192.168.169.131:1031>
   DAG Node: BuildReview
   ...
001 (148.000.000) 04/12 17:40:30 Job executing on host:
<192.168.169.130:1034>
   ...
005 (148.000.000) 04/12 17:40:31 Job terminated.
   (1) Normal termination (return value 0)
   Usr 0 00:00:00, Sys 0 00:00:00 - Run Remote Usage
   Usr 0 00:00:00, Sys 0 00:00:00 - Run Local Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Total Remote Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Total Local Usage
771 - Run Bytes Sent By Job
438 - Run Bytes Received By Job
771 - Total Bytes Sent By Job
438 - Total Bytes Received By Job
...
CalcDebugging.log
000 (150.000.000) 04/12 17:40:42 Job submitted from host:
<192.168.169.131:1031>
DAG Node: CalcMod
...
001 (150.000.000) 04/12 17:40:49 Job executing on host:
<192.168.169.130:1034>
...
005 (150.000.000) 04/12 17:40:50 Job terminated.
(1) Normal termination (return value 0)
Usr 0 00:00:00, Sys 0 00:00:00 - Run Remote Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Run Local Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Total Remote Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Total Local Usage
773 - Run Bytes Sent By Job
438 - Run Bytes Received By Job
773 - Total Bytes Sent By Job
438 - Total Bytes Received By Job
...
CalcTesting.log
000 (154.000.000) 04/12 17:41:01 Job submitted from host:
<192.168.169.131:1031>
DAG Node: CalcTest
...
001 (154.000.000) 04/12 17:41:07 Job executing on host:
<192.168.169.130:1034>
...
005 (154.000.000) 04/12 17:41:08 Job terminated.
(1) Normal termination (return value 0)
Usr 0 00:00:00, Sys 0 00:00:00 - Run Remote Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Run Local Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Total Remote Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Total Local Usage
771 - Run Bytes Sent By Job
438 - Run Bytes Received By Job
771 - Total Bytes Sent By Job
438 - Total Bytes Received By Job
...
DrawDebugging.log

000 (152.000.000) 04/12 17:40:47 Job submitted from host: <192.168.169.131:1031>
   DAG Node: DrawMod

...  

001 (152.000.000) 04/12 17:40:54 Job executing on host: <192.168.169.130:1034>

...  

005 (152.000.000) 04/12 17:40:55 Job terminated.
   (1) Normal termination (return value 0)
   Usr 0 00:00:00, Sys 0 00:00:00 - Run Remote Usage
   Usr 0 00:00:00, Sys 0 00:00:00 - Run Local Usage
   Usr 0 00:00:00, Sys 0 00:00:00 - Total Remote Usage
   Usr 0 00:00:00, Sys 0 00:00:00 - Total Local Usage
   772 - Run Bytes Sent By Job
   438 - Run Bytes Received By Job
   772 - Total Bytes Sent By Job
   438 - Total Bytes Received By Job

...  

DrawTesting.log

000 (156.000.000) 04/12 17:41:12 Job submitted from host: <192.168.169.131:1031>
   DAG Node: DrawTest

...  

001 (156.000.000) 04/12 17:41:27 Job executing on host: <192.168.169.130:1034>

...  

005 (156.000.000) 04/12 17:41:27 Job terminated.
   (1) Normal termination (return value 0)
   Usr 0 00:00:00, Sys 0 00:00:00 - Run Remote Usage
   Usr 0 00:00:00, Sys 0 00:00:00 - Run Local Usage
   Usr 0 00:00:00, Sys 0 00:00:00 - Total Remote Usage
   Usr 0 00:00:00, Sys 0 00:00:00 - Total Local Usage
   771 - Run Bytes Sent By Job
   438 - Run Bytes Received By Job
   771 - Total Bytes Sent By Job
   438 - Total Bytes Received By Job

...  

ImpressDebugging.log

000 (151.000.000) 04/12 17:40:44 Job submitted from host: <192.168.169.131:1031>
   DAG Node: ImpressMod

...
001 (151.000.000) 04/12 17:40:52 Job executing on host: <192.168.169.130:1034>

... 

005 (151.000.000) 04/12 17:40:52 Job terminated.
(1) Normal termination (return value 0)
Usr 0 00:00:00, Sys 0 00:00:00 - Run Remote Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Run Local Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Total Remote Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Total Local Usage
775 - Run Bytes Sent By Job
438 - Run Bytes Received By Job
775 - Total Bytes Sent By Job
438 - Total Bytes Received By Job

... 

ImpressTesting.log

000 (155.000.000) 04/12 17:41:09 Job submitted from host: <192.168.169.131:1031>
DAG Node: ImpressTest

... 

001 (155.000.000) 04/12 17:41:22 Job executing on host: <192.168.169.129:1212>

... 

005 (155.000.000) 04/12 17:41:23 Job terminated.
(1) Normal termination (return value 0)
Usr 0 00:00:00, Sys 0 00:00:00 - Run Remote Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Run Local Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Total Remote Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Total Local Usage
774 - Run Bytes Sent By Job
438 - Run Bytes Received By Job
774 - Total Bytes Sent By Job
438 - Total Bytes Received By Job

... 

WriterDebugging.log

000 (149.000.000) 04/12 17:40:40 Job submitted from host: <192.168.169.131:1031>
DAG Node: WriterMod

... 

001 (149.000.000) 04/12 17:40:47 Job executing on host: <192.168.169.130:1034>

... 

005 (149.000.000) 04/12 17:40:47 Job terminated.
(1) Normal termination (return value 0)
Usr 0 00:00:00, Sys 0 00:00:00 - Run Remote Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Run Local Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Total Remote Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Total Local Usage
774 - Run Bytes Sent By Job
438 - Run Bytes Received By Job
774 - Total Bytes Sent By Job
438 - Total Bytes Received By Job
...

WriterTesting.log

000 (153.000.000) 04/12 17:40:59 Job submitted from host: <192.168.169.131:1031>
   DAG Node: WriterTest
...
001 (153.000.000) 04/12 17:41:17 Job executing on host: <192.168.169.129:1212>
...
005 (153.000.000) 04/12 17:41:19 Job terminated,
   (1) Normal termination (return value 0)
Usr 0 00:00:00, Sys 0 00:00:00 - Run Remote Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Run Local Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Total Remote Usage
Usr 0 00:00:00, Sys 0 00:00:00 - Total Local Usage
773 - Run Bytes Sent By Job
438 - Run Bytes Received By Job
773 - Total Bytes Sent By Job
438 - Total Bytes Received By Job
...


109


[45] University of Wisconsin Madison Computer Science. Condor. the directed acyclic graph manager.  


[49] Jazz Project. The jazz collaborative project.  


[52] Margaret Rouse. Overheard: What the heck is computing in a cloud?  


111